

AVOCET SYSTEMS Z80 ASSEMBLER - VERSION 1.04M SERIAL #00167

TIMEY 2068 STRINGY FLOPPY

PAGE 2

COPYRIGHT (C) 1984 JIM MAYNARD

;**

;** U20 DISASSEMBLED

;**

;** SYSTEM VARIABLES

5C00	ORG 23552		
5C00	KSTATE	DS 8	;USED FOR READ KEYBOARD
5C08	LAST_K	DS 1	;NEW KEY
5C09	REPDEL	DS 1	;REPEAT DELAY
5C0A	REFPR	DS 1	; "
5C0B	DEFADD	DS 2	;ADDRESS OF USER DEFINED FUNCTION
5C0D	K_DATA	DS 1	;COLOR CONTROL FROM THE KEYBOARD
5C0E	TVDATA	DS 2	;AT AND TAB CONTROL
5C10	STRMS	DS 38	;ADDRESS OF CHANNELS ATTACHED TO
5C36	CHARS	DS 2	;CHAR SET-256
5C38	RASP	DS 1	;WARNING BUZZER LEN
5C39	PIN	DS 1	;KEYBOARD CLICK LEN
5C3A	ERR_NR	DS 1	;REPORT NUMBER-1
5C3B	FLAGS	DS 1	
5C3C	TV_FLAG	DS 1	
5C3D	ERR_SP	DS 2	;ADDRESS OF ITEM ON STACK FOR ERROR
5C3E	LIST SP	DS 2	;ADDRESS OF RETURN FROM LISTING
5C41	MODE	DS 1	;WHAT KIND OF CURSOR (KLCEG)
5C42	NEWPPC	DS 2	;NEXT LINE NUM
5C44	NSPPC	DS 1	;WHICH STATEMENT OF NEXT LINE
5C45	PFC	DS 2	;CURRENT LINE
5C47	SUBPFC	DS 1	;CURRENT STATEMENT
5C48	BORDCR	DS 1	;BORDER COLOR*8
5C49	E_PFC	DS 2	;LINE NUMBER W/PRG CURSOR
5C4B	VARS	DS 2	;ADDRESS OF VARIABLES
5C4D	DEST	DS 2	;VARIABLE IN ASSIGNMENT
5C4F	CHANS	DS 2	;CHANNEL DATA
5C51	CURCHL	DS 2	;INFO FOR CURRENT IO
5C53	PROG	DS 2	;BASIC PROGRAM
5C55	NXTLIN	DS 2	;NEXT LINE IN PROGRAM
5C57	DATADD	DS 2	;END OF LAST DATA STATMNT
5C59	E_LIN	DS 2	;COMMAND BEING TYPED
5C5B	K_CUR	DS 2	;CURSOR
5C5D	CH_ADD	DS 2	
5C5F	X_PTR	DS 2	;CHAR AFTER ? MARKER
5C61	WORKSP	DS 2	;TEMP WORKSPACE
5C63	STKBOT	DS 2	;CALCULATOR STACK BOTTOM
5C65	STKEND	DS 2	;TOP
5C67	BREG	DS 1	;CALCULATOR "B" REGISTER
5C68	MEM	DS 2	;ADDRESS OF CALC WORKSPACE
5C6A	ELAGS2	DS 1	;FLAGS (TECH REF)
5C6B	DF_SZ	DS 1	
5C6C	STOP	DS 2	

5C6E	OLDPFC	DS 2	; LINE NUMBER FOR CONTINUE
5C70	OSPCC	DS 1	; WHICH STATEMENT "
5C71	FLAGX	DS 1	;
5C72	STRLEN	DS 2	; LEN OF STRING IN ASSIGNMENT
5C74	T_ADDR	DS 2	; ADDRESS OF NEXT IN SYNTAX TABLE
5C76	SEED	DS 2	; RANDOMIZE SEED
5C78	FRAMES	DS 3	; THREE BYTE FRAME COUNTER
5C7B	UDG	DS 2	; ADDRESS OF FIRST USER DEFINED
5C7D			; GRAPHIC
5C7D	COORDS	DS 2	; X AND Y
5C7F	MISCEL	DS 51	
5CB2	RAMTOP	DS 2	; LAST MEM AVAILABLE TO BASIC
5CB4	P RAMT	DS 2	; LAST PHYSICAL RAM
5CB6	ERRLN	DS 2	; WHICH LINE ON ERROR
5CB8	ERRC	DS 2	; ERROR OCCURED IN LINE NUMBER
5CBA	ERRS	DS 1	; WHICH STATEMENT
5CBB	ERRT	DS 1	; ERROR NUMBER
5CBC	SYSCON	DS 2	; ADDRESS OF CONFIG TABLE
5CBE	MAXBNK	DS 1	; NUMBER OF EXPANSION BANKS IN SYS
5CBF	CURCBN	DS 1	; CURRENT CHANNEL BANK NUMBER
5CC0	MSTBOT	DS 2	; LOCATION ABOVE MACHINE STACK
5CC2	VIDMOD	DS 8	
5CCA	STRNMN	DS 1	; CURRENT STREAM NUMBER

0000		ORG 00H	
0000	XBASIC	EDU \$	; BASE OF EXTENSION ROM
0000 F3		DI	; NO INTERRUPTS
0001 1846		JMPR SKIPO	;
0003 FFFFFFFF		DB 255,255,255,255,255	
0008 2A5D5C	RST8	LHLD CH_ADD	; 5C5D IN LIST ERRR
000B 225F5C		SHLD X_PTR	; 5C5F
000E E1		POP H	
000F 6E		MOV L,M	
0010 FD7500	RST10	MOV 0(Y),L	;
0013 ED7B3D5C		LSPD ERR_SP	; 5C3D
0017 215413		LXI H,1354H	; WHERE TO GO IN HOME ROM
001A E5		PUSH H	
001B 26FF		MVI H,-1	
001D 2E00		MVI L,0	; BANK SELECT
001F E5		PUSH H	
0020 F5		PUSH PSW	
0021 3AC25C		LDA VIDMOD	; 5CC2
0024 A7		ANA A	
0025 00		NOP	
0026 2B04		JRZ SKIP1	
0028 F1		POP PSW	
0029 CD32FD		CALL OFD32H	; THE DISPATCHER IS HERE IF 2 FILES

```

002C      SKIP1      EQU $          ;FOR DISPLAY
002C F1      POP PSW
002D CD7265    CALL 6572H
0030 FFFFFFFF    DB 255,255,255,255,255,255,255,255
0038 F5      PUSH PSW
0039 F3      DI
003A 3AC25C    LDA VIDMOD          ;50C2
003D A7      ANA A
003E 00      NOP
003F 2804      JRZ SKIP2
0041 F1      POP PSW
0042 C36EFA    JMP 0FA6EH
0045      SKIP2      EQU $
0045 F1      POP PSW
0046 C3AE62    JMP 62AEH
0049 3E01      SKIP0      MVI A,1
004B D3F4      OUT 0F4H          ;HORIZONTAL SELECT
004D 1B0B      JMPR SKIP3
004E AF      RAMDAT      XRA A          ;THIS IS MOVED TO RAM AND
                                ;EXECUTED
0050 D3F4      OUT 0F4H          ;HORIZONTAL SELECT
0052      ;ENABLE HOME IN EVERY BANK
0052 D3FF      OUT 0FFH          ;SELECT
0054 11FFFF    LXI D,-1          ;
0057 C3310D    JMP 0D31H          ;EXEC
005A 214F00    SKIP3      LXI H,RAMDAT
005D 110060    LXI D,6000H          ;MOVE HERE
0060 010B00    LXI B,(SKIP3-RAMDAT)
0063 EDB0      LDIR
0065 C30060    JMP 6000H          ;AND SWITCH OUT EXTENSION FROM RAM

;*****
;**
;** TAPE ROUTINES
;**
;*****
0068      TAPE EQU $
08E7      TAPEND EQU TAPE+087FH

;SAVE BYTE ROUTINE
;ENTER WITH DE=LENGTH
;      IX=BASE ADDRESS
;      A=0=HEADER
;      A=0FFH=DATA BLOCK

0068 21E500    SAVBYT      LXI H,0E5H          ;RETURN ADDRESS
0068 E5      PUSH H          ;PUT ON STACK
006C 21801F    LXI H,1F80H          ;FIVE SECOND LEADER FOR HEADER
006F CB7F      BIT 7,A          ;GO AHEAD IF HEADER SAVE

```

0071 2803		JRZ SKIP4	
0073 21980C		LXI H,0C98H	;ELSE TWO SECS FOR DATA BLOCK
0076 08	SKIP4	EXAF	;SAVE FLAG
0077 13		INX D	;LEN=LEN+1
0078 DD2B		DCX X	;BASE ADDRESS=ADDRESS-1 TO ALLOW FOR
	FLAG		
007A F3		DI	
007B 3E02		MVI A,2	;MIC ON AND RED BORDER
007D 47		MOV B,A	
007E 10FE	LOOP0	DJNZ \$	;TIMING LOOP
0080 D3FE		OUT OFEH	;TOGGLE WRITE DATA
0082 EE0F		XRI OFH	
0084 06A4		MVI B,0A4H	;BIT TIME
0086 2D		DCR L	;DEC LOW COUNT
0087 20F5		JRNZ LOOP0	
0089 05		DCR B	;DEC B ALLOW FOR LONGER LOOP
008A 25		DCR H	;DEC THE HIGH COUNTER
008B F27E00		JP LOOP0	;UNTIL LEADER IS COMPLETE
008E 062F		MVI B,02FH	;SEND A SYNC PULSE
0090 10FE		DJNZ \$	
0092 D3FE		OUT OFEH	;MIC ON AND RED
0094 3E0D		MVI A,0DH	
0096 0637		MVI B,037H	
0098 10FE		DJNZ \$	
009A D3FE		OUT OFEH	;OFF AND YELLOW
009C 010E3B		LXI B,03B0EH	;3B=TIMER 0E=MIC OFF AND YELLOW
009F 0B		EXAF	;FLAG (LEADER OR BLOCK)>L FOR SEND
00A0 6F		MOV L,A	
00A1 C3AD00		JMP SAVE	;SAVE START
00A4 7A	SA_LOOP	MOV A,D	;CHECK LENGTH FOR ZERO
00A5 B3		ORA E	
00A6 280C		JRZ SA_PARIT	
00AB DD6E00		MOV L,0(X)	;GET THE NEXT BYTE
00AB 7C	SA_LOFF	MOV A,H	;DO CHECKSUM
00AC AD		XRA L	
00AD 67	SAVE	MOV H,A	
00AE 3E01		MVI A,1	;MIC ON AND BLUE
00B0 37		STC	;SET CARRY
00B1 C3CB00		JMP SAVE_8	;DO EIGHT BITS
00B4 6C	SA_PARIT	MOV L,H	
00B5 18F4		JMPR SA_LOFF	
00B7 79	SABIT2	MOV A,C	
00B8 CB78		BIT 7,B	
00BA 10FE	SABIT1	DJNZ \$	
00BC 3004		JRNC SA_OUT	;IF IF A ZERO
00BE 0642		MVI B,042H	
00C0 10FE		DJNZ \$	;IF A ONE THEN ADD TIME
00C2 D3FE	SA_OUT	OUT OFEH	;TOGGLE WRITE DATA
00C4 063E		MVI B,03EH	
00C6 20EF		JRNZ SABIT2	
00C8 05		DCR B	

00C9 AF		XRA A	
00CA 3C		INR A	
00CB	SAVE_8	EQU \$	
00CB CB15		RALR L	
00CD C2BA00		JNZ SABIT1	
00DD 1B		DCX D	; DEC DE (LENGTH)
00D1 DD23		INX X	; ADVANCE BASE ADDRESS
00D3 0631		MVI B,031H	
00D5 3E7F		MVI A,07FH	
00D7 DBFE		IN OFEH	
00D9 1F		RAR	
00DA D0		RNC	; IF BREAK THEN RETURN
00DB 7A		MOV A,D	
00DC 3C		INR A	; TEST COUNTER FOR FFFF
00DD C2A400		JNZ SA_LOOP	
00E0 063B		MVI B,03BH	; IF ALL GONE RETURN
00E2 10FE		DJNZ \$	; AFTER DELAY
00E4 C9		RET	
00E5 F5	SALDRTN	PUSH PSW	; SAVE CARRY ZXS 053F
00E6 3A485C		LDA BORDCR	; 5C48
00E9 E63B		ANI 03BH	
00EB 0F		RRC	
00EC 0F		RRC	
00ED 0F		RRC	
00EE D3FE		OUT OFEH	; SET BORDER TO NORMAL
00F0 3E7F		MVI A,07FH	; CHECK BREAK KEY
00F2 DBFE		IN OFEH	
00F4 1F		RAR	
00F5 FB		EI	; ENABLE IRQ
00F6 3B02		JRC SALDEND	; IF NO BREAK
00F8 CF	REPORTD	RST 1	; GOTO 0008
00F9 0C		DB 0CH	; ERROR NUMBER
00FA F1	SALDEND	POP PSW	
00FB C9		RET	
00FC 14	LD_BYTES	INR D	; SET ZERO FLAG ZXS 0556
00FD 08		EXAF	; SAVE CCS
00FE 15		DCR D	; RESTORE D
00FF F3		DI	
0100 3E0F		MVI A,0FH	; MAKE BORDER WHITE
0102 D3FE		OUT OFEH	
0104 21E500		LXI H,SALDRTN	; PUT RETURN TO SALDRTN ON STACK
0107 E5		PUSH H	
0108 DBFE		IN OFEH	
010A 1F		RAR	
010B E620		ANI 020H	
010D F602		ORI 02H	
010F 4F		MOV C,A	; PUTS EAR ON/OFF TO C
0110 BF		CMP A	; SETS Z FLAG
0111 C0	LD_BREAK	RNZ	

0112 CD8D01	LD START	CALL LDEDG1	
0115 30FA		JRNC LD_BREAK	
0117 211504		LXI H,0415H	
011A 10FE	LD WAIT	DJNZ LD WAIT	
011C 2B		DCX H	
011D 7C		MOV A,H	; TEST FOR ZERO
011E B5		ORA L	
011F 20F9		JRNZ LD_WAIT	
0121 CD8901		CALL LDEDG2	
0124 30EB		JRNC LD_BREAK	
0126 069C	LD_LEADR	MVI B,09CH	; TIMER
0128 CD8901		CALL LDEDG2	
012B 30E4		JRNC LD_BREAK	
012D 3EC6		MVI A,0C6H	
012F B8		CMP B	
0130 30E0		JRNC LD_START	
0132 24		INR H	
0133 20F1		JRNZ LD_LEADR	; FOR 256 PAIRS
0135 06C9	LD_SYNC	MVI B,0C9H	; TIMING CONSTANT
0137 CD8D01		CALL LDEDG1	
013A 30D5		JRNC LD_BREAK	
013C 7B		MOV A,B	
013D FED4		CPI 0D4H	
013F 30F4		JRNC LD_SYNC	; WAIT FOR SYNC PULSE
0141 CD8D01		CALL LDEDG1	
0144 D0		RNC	; LAST EDGE
	; GET TYPE	FLAG	
0145 79		MOV A,C	
0146 EE03		XRI 03H	
0148 4F		MOV C,A	
0149 2600		MVI H,0	
014B 06B0		MVI B,0B0H	
014D 1B1F		JMPR LD_MARKR	
014F 08	LD_LOOP	EXAF	; RETREIVE FLAGS
0150 2007		JRNZ LD_FLAG	; ONLY FIRST BYTE
0152 300F		JRNC LD_VERIFY	
0154 DD7500		MOV 0(X),L	
0157 1B0F		JMPR LD_NEXT	
0159 CB11	LD_FLAG	RALR C	; SAVE CARRY
015B AD		XRA L	; SEE IF SAME TYPR
015C C0		RNZ	
015D 79		MOV A,C	; RESTORE FLAGS
015E 1F		RAR	
015F 4F		MOV C,A	
0160 13		INX D	
0161 1B07		JMPR LD_DEC	
0163 DD7E00	LD_VERIFY	MOV A,0(X)	; GET FIRST BYTE
0166 AD		XRA L	; COMPARE
0167 C0		RNZ	; RETURN IF <>
0168 DD23	LD_NEXT	INX X	; INCREASE DESTINATION
016A 1B	LD_DEC	DCX D	; DEC COUNT

016B 08		EXAF	;SAVE FLAGS
016C 06B2		MVI B,0B2H	
016E 2E01	LD_MARKR	MVI L,01H	;THIS WILL GO TO CARRY
0170 CD8901	LD_8BITS	CALL LDDEG2	;AFTER EIGHT BITS
0173 D0		RNC	;IF MAX TIME EXCEEDED
0174 3ECB		MVI A,0CBH	;COMPARE TO 3/4 TIME
0176 B8		CMP B	
0177 CB15		RALR L	;CARRY TO DATA
0179 0680		MVI B,0B0H	
017B D27001		JNC LD_8BITS	;TILL WHOLE BYTE
017E 7C		MOV A,H	
017F AD		XRA L	;KEEP CHECKSUM
0180 67		MOV H,A	
0181 7A		MOV A,D	;CHECKSUM SHOULD BE ZERO IF END
0182 B3		ORA E	
0183 20CA		JRNZ LD_LOOP	;JUMP IF COUNTER<>0
0185 7C		MOV A,H	
0186 FE01		CPI 01H	;CARRY SET IF ZERO
0188 C9		RET	;AND RETURN
0189 CD8D01	LDDEG2	CALL LDDEG1	2XS 05E3
018C D0		RNC	
018D 3E16	LDDEG1	MVI A,016H	2XS 05E7
018F 3D	LD_DELAY	DCR A	
0190 20FD		JRNZ LD_DELAY	
0192 A7		ANA A	
0193 04	LD_SAMP	INR B	
0194 CB		RZ	
0195 3E7F		MVI A,07FH	
0197 DBFE		IN 0FEH	
0199 1F		RAR	
019A D0		RNC	;IF BREAK THEN RETURN
019B A9		XRA C	;COMPARE TO LAST EDGE
019C E620		ANI 020H	;MASK
019E 28F3		JRZ LD_SAMP	;IF STILL THE SAME
01A0 79		MOV A,C	;CHANGE LAST LEVEL
01A1 2F		CMA	;COMPLEMENT
01A2 4F		MOV C,A	
01A3 E607		ANI 07H	;BORDER COLOR
01A5 F60B		ORI 0BH	;MIC OFF
01A7 D3FE		OUT 0FEH	
01A9 37		STC	;SET SUCESSFUL OF
01AA C9		RET	;AND RETURN
01AB	SAVEETC	EQU \$	2XS 0605
01AB 3A745C		LDA T_ADDR	;5C74
01AE 01E119		LXI B,019E1H	
01B1 91		SUB C	;TOKEN ADDR-\$E1
01B2 32745C		STA T_ADDR	;5C74
01B5 DDE5		PUSH X	*****
01B7 D9		EXX	
01B8 21EF1B		LXI H,01BEFH	; THIS CALLS
01BB E5		PUSH H	

01BC 2E00	MVI L,0	:
01BE 26FF	MVI H,OFFH	:
01C0 E5	PUSH H	:
01C1 210000	LXI H,0000H	:
01C4 E5	PUSH H	:
01C5 E5	PUSH H	:
01C6 D9	EXX	:
01C7 CD990F	CALL LEAVER	:*****

01CA DDE1	POP X	:
01CC FDCB017E	BIT 7,1(Y)	:
01D0 2866	JRZ SA_DATA	: IF CHECK SYNTAX JMP
01D2 011100	LXI B,011H	: ALLOW 17 LOCATIONS
01D5 3A745C	LDA T_ADDR	: FOR HEADER OF SAVE
01D8 A7	ANA A	:
01D9 2802	JRZ SA_SPACE	:
01DB 0E22	MVI C,022H	: BUT 34 FOR OTHER COMMANDS
01DD DDE5	PUSH X	: CALL SAVE BC SPACES
01DF D9	EXX	: THE REQUIRED AMOUNT OF SPACE
01E0 213000	LXI H,030H	: IS MADE
01E3 E5	PUSH H	:
01E4 2E00	MVI L,0	:
01E6 26FF	MVI H,OFFH	:
01E8 E5	PUSH H	:
01E9 210000	LXI H,0	:
01EC E5	PUSH H	:
01ED E5	PUSH H	:
01EE D9	EXX	:
01EF CD990F	CALL LEAVER	:

01F2 DDE1	POP X	:
01F4 D5	PUSH D	: COPY START ADDRESS TO X-REG
01F5 DDE1	POP X	:
01F7 060B	MVI B,0BH	: ENTER 11 SPACES TO FILE NAME
01F9 3E20	MVI A,20H	:
01FB 12	STAX D	:
01FC 13	INX D	:
01FD 10FC	DJNZ SA_BLNK	:
01FF DD3601FF	MVI 1(X),OFFH	: A NULL NAME IS ONLY FFH
0203 DDE5	PUSH X	: GET THE LENGTH OF THE NAME
0205 D9	EXX	:
0206 21AF2F	LXI H,02FAFH	:
0209 E5	PUSH H	:
020A 2E00	MVI L,0	:
020C 26FF	MVI H,OFFH	:
020E E5	PUSH H	:
020F 210000	LXI H,0	:
0212 E5	PUSH H	:
0213 E5	PUSH H	:

0214 D9		EXX	:
0215 CD990F		CALL LEAVER	:
0218 DDE1		POP X	:
021A 21F6FF		LXI H,-10	;SEE IF MORE THEN 10 LONG
021D 0B		DCX B	:
021E 09		DAD B	:
021F 03		INX B	:
0220 300F		JRNC SA_NAME	;JUMP IF 10 OR LESS
0222 3A745C		LDA T_ADDR	;BUT NO ERROR IF A LOAD TYPE OF
0225 A7		ANA A	;OPERATION IS IN PROGRESS
0226 2002		JRNZ SA_NULL	:
0228 CF	FILNERR	RST 1	;REPORT FILE NAME ERROR
0229 0E		DB 0EH	;ERROR (F)
022A 78	SA_NULL	MOV A,B	;CHECK LENGTH=0
022B B1		ORA C	:
022C 290A		JRZ SA_DATA	;IF NULL NAME
022E 010A00		LXI B,0AH	;ELSE SET LENGTH=10
0231 DDE5	SA_NAME	PUSH X	;START ADDRESS>HL
0233 E1		POP H	:
0234 23		INX H	;SECOND LOCATION
0235 EB		XCHG	;DE...HL
0236 EDB0		LDIR	;COPY THE NAME
0238 DDE5	SA_DATA	PUSH X	;CALL GET CHAR IN HOME ROM
023A D9		EXX	:
023B 211800		LXI H,018H	:
023E E5		PUSH H	:
023F 2E00		MVI L,0	:
0241 26FF		MVI H,0FFH	:
0243 E5		PUSH H	:
0244 210000		LXI H,0	:
0247 E5		PUSH H	:
0248 E5		PUSH H	:
0249 D9		EXX	:
024A CD990F		CALL LEAVER	:
024D DDE1		POP X	:
024F FEE4		CPI 0E4H	;IS THE NEXT CODE THE "DATA" TOKEN
0251 02F202		JNZ SA_SCREE	;JUMP IF NOT
0254 3A745C		LDA T_ADDR	;CHECK MERGE NAME DATA
0257 FE03		CPI 03H	:
0259 CAD908		JZ REPORTC	;ERROR C IF IT IS
025C DDE5		PUSH X	;ELSE GET THE NEXT CHAR
025E D9		EXX	;ADVANCE POINTER
025F 212000		LXI H,020H	;CALL RST20 LOOK FOR VARIABLES
0262 E5		PUSH H	:
0263 2E00		MVI L,0	:
0265 26FF		MVI H,0FFH	:
0267 E5		PUSH H	:
0268 210000		LXI H,0	:
026B E5		PUSH H	:
026C E5		PUSH H	:
026D D9		EXX	:

026E	CD990F	CALL LEAVER	:
0271	D9	EXX	;CALL
0272	21702C	LXI H,02C70H	
0275	E5	PUSH H	
0276	2E00	MVI L,0	
0278	26FF	MVI H,0FFH	
027A	E5	PUSH H	
027B	210000	LXI H,0	
027E	E5	PUSH H	
027F	E5	PUSH H	
0280	D9	EXX	
0281	CD990F	CALL LEAVER	
0284	DDE1	POP X	
0286	CBF9	SET 7,C	;SET BIT 7 OF ARRAY NAME
0288	300B	JRNC SAV_VOLD	;JUMP IF SAVING AN OLD ARRAY
028A	210000	LXI H,0	;USING NEW ARRAY
028D	3A745C	LDA T_ADDR	;THERE WILL BE NO SAVE OR
0290	3D	DCR A	;VERIFY OF NEW ARRAYS
0291	2816	JRZ SAV_VNEW	
0293	CF	RST 1	;ERROR 2
0294	01	DB 1	
0295	C2D908	SAV_VOLD	JNZ REPORTC ;FILE NAME ERR
0298	FDCB017E		BIT 7,1(Y)
029C	2818	JRZ SA_DAT1	;IF CHECKING SYNTAX JUMP
029E	23	INX H	;POINT TO LOW LENGTH
029F	7E	MOV A,M	;LOW LENGTH TO WORK SPACE
02A0	DD770B	MOV 11(X),A	
02A3	23	INX H	;FOLLOWED BY THE HIGH
02A4	7E	MOV A,M	
02A5	DD770C	MOV 12(X),A	;LENGTH
02A8	23	INX H	;STEP PAST THE HIGH BYTE
02A9	DD710E	SAV_VNEW	MOV 14(X),C ;COPY ARRAY NAME TO WORKSPACE
02AC	3E01	MVI A,1	;ASSUME NUMBER ARRAY
02AE	CB71	BIT 6,C	;AND JUMP IF TRUE
02B0	2801	JRZ SAV_TYPE	
02B2	3C	INR A	;MAKE STRING ARRAY TYPE
02B3	DD7700	SAV_TYPE	MOV 0(X),A ;SAVE THE TYPE IN FIRST OF
02B6	EB	SA_DAT1	XCHG ;DE....HL
02B7	DDE5	PUSH X	;CALL NEXT CHAR
02B9	D9	EXX	
02BA	212000	LXI H,020H	
02BD	E5	PUSH H	
02BE	2E00	MVI L,0	
02C0	26FF	MVI H,0FFH	
02C2	E5	PUSH H	
02C3	210000	LXI H,0	
02C6	E5	PUSH H	
02C7	E5	PUSH H	
02C8	D9	EXX	

02C9	CD990F	CALL LEAVER	:
02CC	DDE1	POP X	:
02CE	FE29	CPI 029H	; IS IT A ")"
02D0	20C3	JRNZ SAV_VOLD	; FILE NAME ERROR IF NOT
02D2	DDE5	PUSH X	; ADVANCE CHAR POINTER
02D4	D9	EXX	:
02D5	212000	LXI H,020H	:
02D8	E5	PUSH H	:
02D9	2E00	MVI L,0	:
02DB	26FF	MVI H,0FFH	:
02DD	E5	PUSH H	:
02DE	210000	LXI H,0	:
02E1	E5	PUSH H	:
02E2	E5	PUSH H	:
02E3	D9	EXX	:
02E4	CD990F	CALL LEAVER	:
02E7	DDE1	POP X	:
02E9	FDCB017E	BIT 7,1(Y)	; IF CHECKING SYNTAX THEN RETURN
02ED	CB	RZ	; RETURN IF 0
02EE	EB	XCHG	; ELSE RETURN POINTER TO HL
02EF	C3C904	JMP SA_ALL	:
02F2	FEAA	SA_SCREE CPI 0AAH	; IS IT SCREEN\$ TOKEN
02F4	2038	JRNZ SA_CODE	; IF NOT
02F6	3A745C	LDA T_ADDR	; NOT POSSIBLE TO MERGE
02F9	FE03	CPI 3	:
02FB	CAD908	JZ REPORTC	; ERROR IF MERGE
02FE	DDE5	PUSH X	; CALL NEXT CHAR
0300	D9	EXX	:
0301	212000	LXI H,020H	:
0304	E5	PUSH H	:
0305	2E00	MVI L,0	:
0307	26FF	MVI H,0FFH	:
0309	E5	PUSH H	:
030A	210000	LXI H,0	:
030D	E5	PUSH H	:
030E	E5	PUSH H	:
030F	D9	EXX	:
0310	CD990F	CALL LEAVER	:
0313	DDE1	POP X	:
0315	FDCB017E	BIT 7,1(Y)	; IF SYNTAX CHECK THEN
0319	CB	RZ	; RETURN
031A	DD360B00	MVI 11(X),0	; PASS DISPLAY SIZE TO
031E	DD360C18	MVI 12(X),01BH	; WORKSPACE
0322	210040	LXI H,4000H	; START ADDRESS OF DISPLAY
0325	DD750D	MOV 13(X),L	; ARE ALSO PASSED
0328	DD740E	MOV 14(X),H	:
032B	C34004	JMP SA_TYPE3	:
032E	FEAF	SA_CODE CPI 0AFH	; IS IT CODE TOKEN
0330	C24704	JNZ SA_LINE	:

0333 3A745C	LDA T_ADDR	: NO MERGE-NAME-CODE
0336 FE03	CPI 3	: IF MERGE THEN REPORTC
0338 CAD908	JZ REPORTC	:
033B DDE5	PUSH X	: GET NEXT CHAR
033D D9	EXX	:
033E 212000	LXI H,020H	:
0341 E5	PUSH H	:
0342 2E00	MVI L,0	:
0344 26FF	MVI H,0FFH	:
0346 E5	PUSH H	:
0347 210000	LXI H,0	:
034A E5	PUSH H	:
034B E5	PUSH H	:
034C D9	EXX	:
034D CD990F	CALL LEAVER	:
0350 D9	EXX	: CALL CHECK FOR END OF STATEMENT
0351 21E721	LXI H,021E7H	:
0354 E5	PUSH H	:
0355 2E00	MVI L,0	:
0357 26FF	MVI H,0FFH	:
0359 E5	PUSH H	:
035A 210000	LXI H,0	:
035D E5	PUSH H	:
035E E5	PUSH H	:
035F D9	EXX	:
0360 CD990F	CALL LEAVER	:
0363 DDE1	POP X	:
0365 2020	JRNZ SA_COD1	: IF NOT THEN LOOK FOR START ADDRESS
0367 3A745C	LDA T_ADDR	: NO SAVE-NAME-CODE
036A A7	ANA A	:
036B CAD908	JZ REPORTC	: FILE NAME ERROR
036E DDE5	PUSH X	: ELSE PUT ZERO FOR THE START ADDRESS
0370 D9	EXX	:
0371 21511C	LXI H,1C51H	:
0374 E5	PUSH H	:
0375 2E00	MVI L,0	:
0377 26FF	MVI H,0FFH	:
0379 E5	PUSH H	:
037A 210000	LXI H,0	:
037D E5	PUSH H	:
037E E5	PUSH H	:
037F D9	EXX	:
0380 CD990F	CALL LEAVER	:
0383 DDE1	POP X	:
0385 1835	JMPR SA_COD2	: AND GO FORWARD
0387 DDE5 SA_COD1	PUSH X	: GET THE FIRST NUMBER
0389 D9	EXX	:
038A 21E51B	LXI H,1B51H	:
038D E5	PUSH H	:
038E 2E00	MVI L,0	:
0390 26FF	MVI H,0FFH	:

0392 E5		PUSH H	:
0393 210000		LXI H,0	:
0396 E5		PUSH H	:
0397 E5		PUSH H	:
0398 D9		EXX	:
0399 CD990F		CALL LEAVER	:
039C D9		EXX	: GET THE PRESENT CHAR
039D 211800		LXI H,0018H	:
03A0 E5		PUSH H	:
03A1 2E00		MVI L,0	:
03A3 26FF		MVI H,0FFH	:
03A5 E5		PUSH H	:
03A6 210000		LXI H,0	:
03A9 E5		PUSH H	:
03AA E5		PUSH H	:
03AB D9		EXX	:
03AC CD990F		CALL LEAVER	:
03AF DDE1		POP X	:
03B1 FE2C		CPI 02CH	: IS IT A ","
03B3 2B20		JRZ SA_COD3	: YES GO GET THE LENGTH
03B5 3A745C		LDA T_ADDR	: REPORT ERROR IF NO PARAMETER
03B8 A7		ANA A	: FOR SAVE
03B9 CAD908		JZ REPORTC	:
03BC DDE5	SA_COD2	PUSH X	: PUT A ZERO FOR THE LENGTH
03BE D9		EXX	:
03BF 21511C		LXI H,1C51H	:
03C2 E5		PUSH H	:
03C3 2E00		MVI L,0	:
03C5 26FF		MVI H,0FFH	:
03C7 E5		PUSH H	:
03C8 210000		LXI H,0	:
03CB E5		PUSH H	:
03CC E5		PUSH H	:
03CD D9		EXX	:
03CE CD990F		CALL LEAVER	:
03D1 DDE1		POP X	:
03D3 1B2A		JMER SA_COD4	: AND GO AHEAD
03D5 DDE5	SA_COD3	PUSH X	: ADVANCE PAST COMMA
03D7 D9		EXX	: CALL NEXT CHAR
03D8 212000		LXI H,020H	:
03DB E5		PUSH H	:
03DC 2E00		MVI L,0	:
03DE 26FF		MVI H,0FFH	:
03E0 E5		PUSH H	:
03E1 210000		LXI H,0	:
03E4 E5		PUSH H	:
03E5 E5		PUSH H	:
03E6 D9		EXX	:
03E7 CD990F		CALL LEAVER	:
03EA D9		EXX	: NOW FETCH THE LENGTH
03EB 21E51B		LXI H,1BE5H	:

03EE E5	PUSH H	:	
03EF 2E00	MVI L,0	:	
03F1 26FF	MVI H,OFFH	:	
03F3 E5	PUSH H	:	
03F4 210000	LXI H,0	:	
03F7 E5	PUSH H	:	
03F8 E5	PUSH H	:	
03F9 D9	EXX	:	
03FA CD990F	CALL LEAVER	:	
03FD DDE1	POP X	:	
03FF FDCB017E SA_COD4	BIT 7,1(Y)	:	; IF SYNTAX THEN RETURN
0403 C8	RZ	:	
0404 DDE5	PUSH X	:	
0406 D9	EXX	:	; MAKE LENGTH TO INT-BC
0407 21231F	LXI H,1F23H	:	
040A E5	PUSH H	:	
040B 2E00	MVI L,0	:	
040D 26FF	MVI H,OFFH	:	
040F E5	PUSH H	:	
0410 210000	LXI H,0	:	
0413 E5	PUSH H	:	
0414 E5	PUSH H	:	
0415 D9	EXX	:	
0416 CD990F	CALL LEAVER	:	
0419 DDE1	POP X	:	
041B DD710B	MOV 11(X),C	:	; AND MOVE TO HEADER AREA
041E DD700C	MOV 12(X),B	:	
0421 DDE5	PUSH X	:	; NOW THE START ADDRESS
0423 D9	EXX	:	
0424 21231F	LXI H,1F23H	:	
0427 E5	PUSH H	:	
0428 2E00	MVI L,0	:	
042A 26FF	MVI H,OFFH	:	
042C E5	PUSH H	:	
042D 210000	LXI H,0	:	
0430 E5	PUSH H	:	
0431 E5	PUSH H	:	
0432 D9	EXX	:	
0433 CD990F	CALL LEAVER	:	
0436 DDE1	POP X	:	
0438 DD710D	MOV 13(X),C	:	; START TO HEADER
043B DD700E	MOV 14(X),B	:	
043E 60	MOV H,B	:	; MOVE POINTER TO HL
043F 69	MOV L,C	:	
0440 DD360003 SA_TYPE3	MVI 0(X),3	:	; ENTER THE TYPE NUMBER
0444 C3C904	JMP SA_ALL	:	
0447 FECA SA_LINE	CPI OCAH	:	; IS IT "LINE" TOKEN
0449 280B	JRZ SA_LIN1	:	; IF IT IS
044B FDCB017E	BIT 7,1(Y)	:	; RETURN IF SYNTAX CHECK
044F C8	RZ	:	

0450 DD360E80	MVI 14(X),080H	;AN 80 IF END OF HEADER
0454 1853	JMPR SA_TYPO	
0456 3A745C	SA_LIN1 LDA T_ADDR	;ONLY SAVE ALLOWED
0459 A7	ANA A	;IF NOT
045A C2D90B	JNZ REPORTC	;THEN ERROR
045D DDE5	PUSH X	;GET NEXT CHAR
045F D9	EXX	;TO POINT TO NUMBER
0460 212000	LXI H,020H	
0463 E5	PUSH H	
0464 2E00	MVI L,0	
0466 26FF	MVI H,OFFH	
0468 E5	PUSH H	
0469 210000	LXI H,0	
046C E5	PUSH H	
046D E5	PUSH H	
046E D9	EXX	
046F CD990F	CALL LEAVER	
0472 D9	EXX	;GET NUMBER
0473 21E51B	LXI H,1BE5H	
0476 E5	PUSH H	
0477 2E00	MVI L,0	
0479 26FF	MVI H,OFFH	
047B E5	PUSH H	
047C 210000	LXI H,0	
047F E5	PUSH H	
0480 E5	PUSH H	
0481 D9	EXX	
0482 CD990F	CALL LEAVER	
0485 DDE1	POP X	
0487 FDCB017E	BIT 7,1(Y)	;IF SYNTAX THEN RETURN
048B C8	RZ	
048C DDE5	PUSH X	;MAKE INTEGER IN BC
048E D9	EXX	
048F 21231F	LXI H,1F23H	
0492 E5	PUSH H	
0493 2E00	MVI L,0	
0495 26FF	MVI H,OFFH	
0497 E5	PUSH H	
0498 210000	LXI H,0	
049B E5	PUSH H	
049C E5	PUSH H	
049D D9	EXX	
049E CD990F	CALL LEAVER	
04A1 DDE1	POP X	
04A3 DD710D	MOV 13(X),C	;AND STORE IN HEADER
04A6 DD700E	MOV 14(X),B	
04A9 DD360000	SA_TYPO MVI 0(X),0	;MARK TYPE
04AD 2A595C	LHLD E_LIN	;END OF VARIABLE AREA
04B0 ED5B535C	LDED PROG	;START OF BASIC PROGRAM
04B4 37	STC	;SUBTRACT

04B5 ED52	DSBC D	
04B7 DD750B	MOV 11(X),L	;= PROGRAM + VARIABLES
04BA DD740C	MOV 12(X),H	;
04BD 2A4B5C	LHLD VARS	;THIS FOR PROGRAM ONLY
04C0 ED52	DSBC D	
04C2 DD750F	MOV 15(X),L	
04C5 DD7410	MOV 16(X),H	
04C8 EB	XCHG	; POINTER > HL

```

;*****
; **
; ** HEADERS COMPLETE
; ** 00 - TYPE
; ** 01 - 0A -NAME ELSE 01=FF IF NO NAME
; ** 0B - 0C - LENGTH OF BLOCK
; ** 0D - 10 - OTHER PARAMETERS DEPEND ON TYPE
; **
;*****

```

04C9 3A745C	SA ALL	LDA T ADDR	;SEPARATE SAVE FROM ELSE
04CC A7		ANA A	
04CD CA510B		JZ SAVCNT	;SAVE GO AHEAD
04D0 E5		PUSH H	;SAVE HEADER POINTER
04D1 011100		LXI B,011H	;MAKE NEW ONE TO PUT
04D4 DD09		DADX B	;HEADER FROM THE TAPE AHEAD
04D6 DDE5	LD_LOOK	PUSH X	;LOOP TILL HEADER IS READ
04D8 111100		LXI D,011H	;FROM TAPE (LEN=17)
04DB AF		XRA A	;SIGNAL GET HEADER
04DC 37		STC	;
04DD CD5B16		CALL LODHDR	;CHECK FOR STRINGY HEADER
04E0 DDE1		POP X	;
04E2 30F2		JRNC LD_LOOK	;IF ERROR THEN LOOP
04E4 3EFE		MVI A,0FEH	;CHANNEL 'S' OPEN
04E6 DDE5		PUSH X	;
04E8 D9		EXX	;
04E9 213012		LXI H,1230H	;CALL CHANELL OPEN
04EC E5		PUSH H	;
04ED 2E00		MVI L,0	;
04EF 26FF		MVI H,0FFH	;
04F1 E5		PUSH H	;
04F2 210000		LXI H,0	;
04F5 E5		PUSH H	;
04F6 E5		PUSH H	;
04F7 D9		EXX	;
04F8 CD990F		CALL LEAVER	;
04FB DDE1		POP X	;
04FD FD365203		MVI 52H(Y),3	;SET SCROLL COUNTER
0501 0E80		MVI C,0B0H	;NO-MATCH FLAG
0503 DD7E00		MOV A,0(X)	;GET TYPE

0506 DBEEF		CMP -17(X)	;CMP TO DESIRED TYPE
0509 2002		JRNZ LD_TYPE	; IF NO MATCH
050B 0EF6		MVI C,0F6H	;ELSE SAY -10 TO MATCH
050D EE04	LD_TYPE	CPI 4	:
050F 30C5		JRNC LD_LOOK	; IF FOUR OR MORE THEN GET ANOTHER
0511 11A83C		LXI D,3CASH	;THE BASE OF MESSAGE BLOCK
0514 C5		PUSH B	;SAVE C
0515 DDE5		PUSH X	;START THE MESSAGE
0517 D9		EXX	:
0518 213F07		LXI H,073FH	:
051B E5		PUSH H	:
051C 2E00		MVI L,0	:
051E 26FF		MVI H,0FFH	:
0520 E5		PUSH H	:
0521 210000		LXI H,0	:
0524 E5		PUSH H	:
0525 E5		PUSH H	:
0526 D9		EXX	:
0527 CD990F		CALL LEAVER	:
052A DDE1		POP X	:
052C C1		POP B	;RESTORE B
052D DDE5		PUSH X	;MAKE D=NEW NAME FROM TAPE
052F D1		POP D	:
0530 21FOFF		LXI H,0FFFOH	; H=OLD NAME
0533 19		DAD D	:
0534 060A		MVI B,10	;TEN CHARS TO BE COMPARED
0536 7E		MOV A,M	; IF NOT A NULL NAME
0537 3C		INR A	:
0538 2003		JRNZ LD_NAME	; IF REAL NAME JMP
053A 79		MOV A,C	;ELSE SET 10 COMPARED FLAG
053B 80		ADD B	:
053C 4F		MOV C,A	:
053D 13	LD_NAME	INX D	;ADVANCE TO NEXT CHAR NEW
053E 1A		LDAX D	;AND
053F BE		CMP M	;COMPARE TO OLD
0540 23		INX H	:
0541 2001		JRNZ PRTCHR	;DONT COUNT IF NOT=
0543 0C		INR C	;ELSE COUNT
0544 DDE5	PRTCHR	PUSH X	;PRINT THE NEW CHAR
0546 D9		EXX	;CALL PRINT CHAR
0547 211000		LXI H,10H	:
054A E5		PUSH H	:
054B 2E00		MVI L,0	:
054D 26FF		MVI H,0FFH	:
054F E5		PUSH H	:
0550 210000		LXI H,0	:
0553 E5		PUSH H	:
0554 E5		PUSH H	:
0555 D9		EXX	:
0556 CD990F		CALL LEAVER	:
0559 DDE1		POP X	:

055B 10E0	DJNZ LD NAME	; LOOP FOR TEN
055D CB79	BIT 7,C	; READ NEW HEADER FROM TAPE IF <>
055F C2D604	JNZ LD_LOOK	;
0562 3E0D	MVI A,13	; ELSE SEND A CARRIAGE RETURN
0564 DDES	PUSH X	; CALL PRINT CHAR
0566 D9	EXX	;
0567 211000	LXI H,010H	;
056A E5	PUSH H	;
056B 2E00	MVI L,0	;
056D 26FF	MVI H,OFFH	;
056F E5	PUSH H	;
0570 210000	LXI H,0	;
0573 E5	PUSH H	;
0574 E5	PUSH H	;
0575 D9	EXX	;
0576 CD990F	CALL LEAVER	;
0579 DDE1	POP X	;
057B E1	POP H	; RESTORE POINTER AND SEPARATE INTO
057C DD7E00	MOV A,0(X)	; VERIFY, LOAD, OR MERGE
057F FE03	CPI 3	; CODE AND SCREEN* VERIFY
0581 280C	JRZ VRCNT	;
0583 3A745C	LDA T_ADDR	;
0586 3D	DCR A	; IS IT LOAD
0587 CACC05	JZ LDCNT	; JUMP IF YES
058A FE02	CPI 2	; IS IT MERGE
058C CAE506	JZ MECNT	; YES JMP
058F E5	PUSH H	; SAVE THE POINTER
0590 DD6EFA	MOV L,-6(X)	; GET BYTE COUNT
0593 DD66FB	MOV H,-5(X)	;
0596 DD5E0B	MOV E,11(X)	; GET NUMBER FROM NEW HEADER
0599 DD560C	MOV D,12(X)	;
059C 7C	MOV A,H	; IF COUNT = 0 THEN GO
059D B5	ORA L	;
059E 280D	JRZ VRCNT1	;
05A0 ED52	DSBC D	; THE COUNTS MUST HAVE = LEN
05A2 3826	JRC REPORTR	; ELSE ERR R LOAD
05A4 2807	JRZ VRCNT1	; ACCEPT = LEN
05A6 DD7E00	MOV A,0(X)	;
05A9 FE03	CPI 3	; ALSO IF VERIFY
05AB 201D	JRNZ REPORTR	;
05AD E1	POP H	; GET START TO USE
05AE 7C	MOV A,H	; IF STAQRT=0 THEN USE NEW START
05AF B5	ORA L	;
05B0 2006	JRNZ VRCNT2	;
05B2 DD6E0D	MOV L,13(X)	; ELSE GET NEW ONE
05B5 DD660E	MOV H,14(X)	;
05B8 E5	PUSH H	; POINTER TO X
05B9 DDE1	POP X	;
05BB 3A745C	LDA T_ADDR	; CHECK FOR LOAD
05BE FE02	CPI 2	;
05C0 37	STC	; CARRY FLAG = LOAD

05C1 2001	JRNZ VRCNT3	
05C3 A7	ANA A	;SET VERIFY = NO CARRY
05C4 3EFF	VRCNT3 MVI A,OFFH	;ACCEPT DATA BLOCK ONLY
05C6 CDAE16	LD_BLK CALL ESELOAD	;LOAD/VERIFY DATA BLOCK
05C9 DB	RC	;RETURN IF NO ERROR
05CA CF	REPORTR RST 1	;REPORT ERROR R
05CB 1A	DB 01AH	;AND RETRUN
05CC DD5E0B	LDCNT MOV E,11(X)	;GET NUMBER IN NEW HEADER 2x5 0808
05CE DD560C	MOV D,12(X)	
05D2 E5	PUSH H	;SAVE DESTINATION
05D3 7C	MOV A,H	;IF LOADING A EXISTING ARRAY
05D4 B5	ORA L	;THEN JUMP
05D5 2006	JRNZ LDCNT1	
05D7 13	INX D	;ELSE INCREASE LEN BY 3
05D8 13	INX D	;FOR NAME-AND LENGTH
05D9 13	INX D	;OF NEW VARIABLE
05DA EB	XCHG	; DE...HL
05DB 180C	JMPR LDCNT2	
05DD DD6EFA	LDCNT1 MOV L,-6(X)	;GET SIZE OF EXISTING
05E0 DD66FB	MOV H,-5(X)	;ARRAY OR PROG+VARS
05E3 EB	XCHG	;GO FORWARD IF NO MORE
05E4 37	STC	;ROOM IS REQUIRED
05E5 ED52	DSBC D	
05E7 391D	JRC LD_DATA	
05E9 110500	LDCNT2 LXI D,5	;FIVE BYTE OVER
05EC 19	DAD D	
05ED 44	MOV B,H	;MOVE TO BC
05EE 4D	MOV C,L	
05EF DDE5	PUSH X	;CALL TEST FOR ROOM
05F1 D9	EXX	
05F2 21BB1F	LXI H,1FBBH	
05F5 E5	PUSH H	
05F6 2E00	MVI L,0	
05F8 26FF	MVI H,OFFH	
05FA E5	PUSH H	
05FB 210000	LXI H,0	
05FE E5	PUSH H	
05FF E5	PUSH H	
0600 D9	EXX	
0601 CD990F	CALL LEAVER	
0604 DDE1	POP X	
0606 E1	LD_DATA POP H	;GET POINTER
0607 DD7E00	MOV A,0(X)	;IF LOAD PROGRAM GO-AHEAD
060A A7	ANA A	
060B 2B66	JRZ LD PROG	
060D 7C	MOV A,H	;GO FORWARD IF LOADING A NEW ARRAY
060E B5	ORA L	
060F 2B27	JRZ LD DAT1	
0611 2B	DCX H	;ELSE THE LENGTH BOF THE EXISTING
0612 46	MOV B,M	;VARIABLE

0613 2B	DCX H	:FROM THE VARIABLE AREA
0614 4E	MOV C,M	:
0615 2B	DCX H	:NOW THE NAME
0616 03	INX B	:+3 FOR THE LEN ETC
0617 03	INX B	:
0618 03	INX B	:
0619 DD225F5C	SIXD X PTR	:SAVE IX WHILE RECLAIMING
061D DDE5	PUSH X	:OR DELETING THE EXISTING VARIABLE
061F D9	EXX	:
0620 215017	LXI H,1750H	:
0623 E5	PUSH H	:
0624 2E00	MVI L,0	:
0626 26FF	MVI H,0FFH	:
0628 E5	PUSH H	:
0629 210000	LXI H,0	:
062C E5	PUSH H	:
062D E5	PUSH H	:
062E D9	EXX	:
062F CD990F	CALL LEAVER	:
0632 DDE1	POP X	:
0634 DD2A5F5C	LIXD X_PTR	:
0638 2A595C LD DAT1	LHLD E LIN	:THIS IS THE END OF THE VARIABLES
063B 2B	DCX H	:MARKER (80H)
063C DD4E0B	MOV C,11(X)	:GET LEN OF NEW ARRAY
063F DD460C	MOV B,12(X)	:
0642 C5	PUSH B	:SAVE
0643 03	INX B	:ADD THREE FOR THE NAME AND LEN
0644 03	INX B	:
0645 03	INX B	:
0646 DD7EFD	MOV A,-3(X)	:NEW NAME
0649 F5	PUSH PSW	:SAVE IT
064A DDE5	PUSH X	:CALL MAKE ROOM FOR VARIABLE
064C D9	EXX	:
064D 21BB12	LXI H,12BBH	:
0650 E5	PUSH H	:
0651 2E00	MVI L,0	:
0653 26FF	MVI H,0FFH	:
0655 E5	PUSH H	:
0656 210000	LXI H,0	:
0659 E5	PUSH H	:
065A E5	PUSH H	:
065B D9	EXX	:
065C CD990F	CALL LEAVER	:
065F DDE1	POP X	:
0661 23	INX H	:ONE PAST THE END
0662 F1	POP PSW	:SAVE THE NAME TO
0663 77	MOV M,A	:AREA
0664 D1	POP D	:AND THE LEN
0665 23	INX H	:
0666 73	MOV M,E	:
0667 23	INX H	:

0668 72	MOV M,D	:
0669 23	INX H	; HL POINTS TO FIRST BYTE
066A E5	PUSH H	; TO BE FILLED FROM TAPE
066B DDE1	POP X	; HL-IX
066D 37	STC	; LOAD FLAG
066E 3EFF	MVI A,OFFH	; LOAD DATA BLOCK FLAG
0670 C3C605	JMP LD_BLK	
0673 EB	LD_PROG XCHG	; SAVE DESTINATION FLAG
0674 2A595C	LHLD E_LIN	; GET END OF VARS MARKER
0677 2B	DCX H	:
0678 DD225F5C	SIXD X_PTR	:
067C DD4E0B	MOV C,11(X)	; GET LEN OF NEW BLOCK
067F DD460C	MOV B,12(X)	:
0682 C5	PUSH B	; SAVE IT AND
0683 DDE5	PUSH X	; DELETE THE OLD PROGRAM+VARS
0685 D9	EXX	; CALL RECLAIM
0686 214D17	LXI H,174DH	:
0689 E5	PUSH H	:
068A 2E00	MVI L,0	:
068C 26FF	MVI H,OFFH	:
068E E5	PUSH H	:
068F 210000	LXI H,0	:
0692 E5	PUSH H	:
0693 E5	PUSH H	:
0694 D9	EXX	:
0695 CD990F	CALL LEAVER	:
0698 DDE1	POP X	:
069A C1	POP B	; SAVE NEW POINTER
069B E5	PUSH H	; AND LEN
069C C5	PUSH B	:
069D DDE5	PUSH X	; CALL MAKE ROOM
069F D9	EXX	:
06A0 21BB12	LXI H,12BBH	:
06A3 E5	PUSH H	:
06A4 2E00	MVI L,0	:
06A6 26FF	MVI H,OFFH	:
06A8 E5	PUSH H	:
06A9 210000	LXI H,0	:
06AC E5	PUSH H	:
06AD E5	PUSH H	:
06AE D9	EXX	:
06AF CD990F	CALL LEAVER	:
06B2 DDE1	POP X	:
06B4 DD2A5F5C	LIXD X_PTR	:
06B8 23	INX H	:
06B9 DD4E0F	MOV C,15(X)	; SET THE VARIABLE START POINTER
06BC DD4610	MOV B,16(X)	:
06BE 09	DAD B	:
06C0 224B5C	SHLD VARS	:
06C3 DD660E	MOV H,14(X)	; CHECK FOR LINE NUMBER

06C6 7C		MOV A,H	:	
06C7 E6C0		ANI 0C0H	:	
06C9 200A		JRNZ LD_PRG1	:	; IF NONE
06CB DD4E0D		MOV L,13(X)	:	; ELSE
06CE 22425C		SHLD NEWPPC	:	; SET VARIABLES
06D1 FD360A00		MVI 10(Y),0	:	
06D5 D1	LD_PRG1	POP D	:	; RESTORE LEN
06D6 DDE1		POP X	:	; AND STARTING ADDRESS
06D8 37		STC	:	; SET LOAD FLAG
06D9 3EFF		MVI A,OFFH	:	; SET DATA BLOCK FLAG
06DB 2A535C		LHLD PROG	:	; WHERE TO PUT IT
06DE 2B		DCX H	:	
06DF 22575C		SHLD DATADD	:	
06E2 C3C605		JMP LD_BLK	:	; AND LOAD IT
06E5 DD4E0B	MECNT	MOV C,11(X)	:	; GET BTHE LEN ZXS 08B6
06E8 DD460C		MOV B,12(X)	:	
06EB C5		PUSH B	:	; COPY LEN
06EC 03		INX B	:	; MAKE A SPACE IN RAM
06ED DDE5		PUSH X	:	; CALL MAKE ROOM
06EF D9		EXX	:	
06F0 213000		LXI H,0030H	:	
06F3 E5		PUSH H	:	
06F4 2E00		MVI L,0	:	
06F6 26FF		MVI H,OFFH	:	
06F8 E5		PUSH H	:	
06F9 210000		LXI H,0	:	
06FC E5		PUSH H	:	
06FD E5		PUSH H	:	
06FE D9		EXX	:	
06FF CD990F		CALL LEAVER	:	
0702 DDE1		POP X	:	
0704 3680		MVI M,80H	:	; PUT THE END MARKER
0706 EB		XCHG	:	; DE....HL
0707 D1		POP D	:	; GET LEN
0708 E5		PUSH H	:	; PUSH START
0709 E5		PUSH H	:	; START-IX
070A DDE1		POP X	:	
070C 37		STC	:	; SET LOAD FLAG
070D 3EFF		MVI A,OFFH	:	; SET DATA BLOCK FLAG
070F CDC605		CALL LD_BLK	:	
0712 E1		POP H	:	; START OF NEW PROG FOR MERGE
0713 ED5B535C		LDED PROG	:	; START OF NEW
0717 7E	MENEWLP	MOV A,M	:	; CHECK LINE NUMBER FOR END
0718 E6C0		ANI 0C0H	:	
071A 202D		JRNZ MEVARLP	:	; JUMP IF DONE
071C 1A	MEOLDLP	LDAX D	:	; GET HIGH LINE NUMBER
071D 13		INX D	:	; AND COMPARE
071E 9E		CMP M	:	; TO (HL)
071F 23		INX H	:	
0720 2002		JRNZ MEOLD1	:	; IF NOT =

0722 1A		LDAX D	; ELSE TRY LOW
0723 BE		CMP M	
0724 1B	MEOLD1	DCX D	; BACK THE LINE POINTERS
0725 2B		DCX H	
0726 301C		JRNC MENEWL2	; JUMP IF GOOD PLACE TO PUT LINE
0728 E5		PUSH H	; ELSE GET ADDRESS OF NEXT OLD LINE
0729 EB		XCHG	; DE...HL
072A DDE5		PUSH X	; CALL GET NEXT LINE
072C D9		EXX	
072D 212017		LXI H,1720H	
0730 E5		PUSH H	
0731 2E00		MVI L,0	
0733 26FF		MVI H,0FFH	
0735 E5		PUSH H	
0736 210000		LXI H,0	
0739 E5		PUSH H	
073A E5		PUSH H	
073B D9		EXX	
073C CD990F		CALL LEAVER	
073F DDE1		POP X	
0741 E1		POP H	; AND LOOP FOR EACH OF OLD LINES
0742 18D8		JMPR MEOLDLP	
0744 CD9907	MENEWL2	CALL MENTER	; ENTER LINE AND LOOP
0747 18CE		JMPR MENEWL2	; CE
0749 7E	MEVARLP	MOV A,M	; NOW MERGE VARIABLES
074A 4F		MOV C,A	; TEST FOR END (80H)
074B FE80		CPI 80H	
074D C8		RZ	; RETURN IF END
074E E5		PUSH H	; SAVE NEW POINTER
074F 2A4B5C		LHLD VARS	; GET OLD VARS
0752 7E	MEOLDVP	MOV A,M	; GET EACH NAME AND TEST
0753 FE80		CPI 80H	; FOR END
0755 2839		JRZ MEVARL2	; JUMP ON END
0757 B9		CMP C	; COMPARE NAMES
0758 281C		JRZ MEOLDV2	; IF THE SAME
075A C5	MEOLDV1	PUSH B	; SAVE THE NEW VARIABLE NAME
075B DDE5		PUSH X	; AND LOCATE NEXT OLD ONE
075D D9		EXX	; CALL NEXT ONE
075E 212017		LXI H,1720H	
0761 E5		PUSH H	
0762 2E00		MVI L,0	
0764 26FF		MVI H,0FFH	
0766 E5		PUSH H	
0767 210000		LXI H,0	
076A E5		PUSH H	
076B E5		PUSH H	
076C D9		EXX	
076D CD990F		CALL LEAVER	
0770 DDE1		POP X	
0772 C1		POP B	; RESTORE POINTER
0773 EB		XCHG	

0774 18DC		JMPL MEOLDV2	; AND LOOP
0776 E6E0	MEOLDV2	ANI 0E0H	; GET BITS 5,6,7
0778 FEA0		CPI 0A0H	; ALL VARIABLES EXCEPT LONG
077A 2012		JRNZ MEVARL1	; NAMES
077C D1		POP D	; FIRST CHAR OF NEW
077D D5		PUSH D	
077E E5		PUSH H	; POINTER TO NEW
077F 23	MEOLDV3	INX H	; INC OLD AND NEW
0780 13		INX D	
0781 1A		LDAX D	; COMPARE LONG NAMES
0782 BE		CMP M	
0783 2006		JRNZ MEOLDV4	; IF <>
0785 17		RAL	; GO TILL LAST CHAR IS FOUND
0786 30F7		JRNC MEOLDV3	
0788 E1		POP H	; OLD NAME
0789 1803		JMPL MEVARL1	; =====
078B E1	MEOLDV4	POP H	; RESTORE AND
078C 18CC		JMPL MEOLDV1	; LOOP <><>
078E 3EFF	MEVARL1	MVI A, 0FFH	; REPLACE FLAG
0790 D1	MEVARL2	POP D	; NEW NAME
0791 EB		XCHG	
0792 3C		INR A	; ZERO FLAG SET=REPLACE
0793 37		STC	; DO VARIABLE
0794 CD9907		CALL MENTER	
0797 18B0		JMPL MEVARL2	; TRY NEXT LOOP
0799 2034	MENTER	JRNZ MENT1	; FOR ADDITION
079B 08		EXAF	; SAVE FLAGS
079C 225F5C		SHLD X PTR	; SAVE NEW POINTER WHILE OLD
079F EB		XCHG	; IS REPLACED
07A0 DDE5		PUSH X	; CALL GET NEXT ONE
07A2 D9		EXX	
07A3 212017		LXI H, 1720H	
07A6 E5		PUSH H	
07A7 2E00		MVI L, 0	
07A9 26FF		MVI H, 0FFH	
07AB E5		PUSH H	
07AC 210000		LXI H, 0	
07AF E5		PUSH H	
07B0 E5		PUSH H	
07B1 D9		EXX	
07B2 CD990F		CALL LEAVER	
07B5 D9		EXX	; CALL RECLAIM
07B6 215017		LXI H, 1750H	
07B9 E5		PUSH H	
07BA 2E00		MVI L, 0	
07BC 26FF		MVI H, 0FFH	
07BE E5		PUSH H	
07BF 210000		LXI H, 0	
07C2 E5		PUSH H	
07C3 E5		PUSH H	
07C4 D9		EXX	

07C5	CD990F		CALL LEAVER	:
07C8	DDE1		POP X	:
07CA	EB		XCHG	:
07CB	2A5F5C		LHLD X_PTR	:
07CE	08		EXAF	; RESTORE FLAGS
07CF	08	MENT1	EXAF	; ADD THE NEW ONE
07D0	05		PUSH D	; SAVE DESTINATION
07D1	DDE5		PUSH X	:
07D3	D9		EXX	; CALL NEXT ONE
07D4	212017		LXI H, 1720H	:
07D7	E5		PUSH H	:
07D8	2E00		MVI L, 0	:
07DA	26FF		MVI H, 0FFH	:
07DC	E5		PUSH H	:
07DD	210000		LXI H, 0	:
07E0	E5		PUSH H	:
07E1	E5		PUSH H	:
07E2	D9		EXX	:
07E3	CD990F		CALL LEAVER	:
07E6	DDE1		POP X	:
07E8	225F5C		SHLD X_PTR	:
07EB	2A535C		LHLD PROG	; SAVE PROG
07EE	E3		DB 0E3H	; EX (HL) SP
07EF	C5		PUSH B	; SAVE LEN
07F0	08		EXAF	; RESTORE FLAGS
07F1	381B		JRC MENT2	; IF ADDING VARIABLE JMP
07F3	2B		DCX H	; ELSE ADD NEW LINE
07F4	DDE5		PUSH X	; CALL MAKE ROOM
07F6	D9		EXX	:
07F7	21BB12		LXI H, 12BBH	:
07FA	E5		PUSH H	:
07FB	2E00		MVI L, 0	:
07FD	26FF		MVI H, 0FFH	:
07FE	E5		PUSH H	:
0800	210000		LXI H, 0	:
0803	E5		PUSH H	:
0804	E5		PUSH H	:
0805	D9		EXX	:
0806	CD990F		CALL LEAVER	:
0809	DDE1		POP X	:
080B	23		INX H	:
080C	1817		JMPR MENT3	:
080E	DDE5	MENT2	PUSH X	; CALL MAKE ROOM
0810	D9		EXX	:
0811	21BB12		LXI H, 12BBH	:
0814	E5		PUSH H	:
0815	2E00		MVI L, 0	:
0817	26FF		MVI H, 0FFH	:
0819	E5		PUSH H	:
081A	210000		LXI H, 0	:
081D	E5		PUSH H	:

COPYRIGHT (C) 1984 JIM MAYNARD

081E E5		PUSH H	:
081F D9		EXX	:
0820 CD990F		CALL LEAVER	:
0823 DDE1		POP X	:
0825 23	MENT3	INX H	: NEW LOCATION
0826 C1		POP B	: LENGTH
0827 D1		POP D	: RESTORE PROG
0828 ED53535C		SDED PROG	:
082C ED5B5F5C		LDED X_PTR	:
0830 C5		PUSH B	: SAVE AGIN
0831 D5		PUSH D	:
0832 EB		XCHG	: AND COPY
0833 EDB0		LDIR	:
0835 E1		POP H	: REMOVE NEW VAR/LINE FROM WORK SPC
0836 C1		POP B	:
0837 D5		PUSH D	:
0838 DDE5		PUSH X	:
083A D9		EXX	: CALL RECLAIM
083B 215017		LXI H,1750H	:
083E E5		PUSH H	:
083F 2E00		MVI L,0	:
0841 26FF		MVI H,0FFH	:
0843 E5		PUSH H	:
0844 210000		LXI H,0	:
0847 E5		PUSH H	:
0848 E5		PUSH H	:
0849 D9		EXX	:
084A CD990F		CALL LEAVER	:
084D DDE1		POP X	:
084F D1		POP D	: AND RETURN W/OLD POINTER
0850 C9		RET	:
0851 E5	SAVCNT	PUSH H	: SAVE POINTER ZXS 0970
0852 3EFD		MVI A,0FDH	: OPEN CHANNEL "K"
0854 DDE5		PUSH X	: CALL CHANOPEN
0856 D9		EXX	:
0857 213012		LXI H,1230H	:
085A E5		PUSH H	:
085B 2E00		MVI L,0	:
085D 26FF		MVI H,0FFH	:
085F E5		PUSH H	:
0860 210000		LXI H,0	:
0863 E5		PUSH H	:
0864 E5		PUSH H	:
0865 D9		EXX	:
0866 CD990F		CALL LEAVER	:
0869 DDE1		POP X	:
086B AF		XRA A	: SEND FIRST MESSAGE
		LXI D,3C89H	:
086C CD4D18		CALL GETMSG	: IF ESF SAVE WONT COME BACK
086F DDE5		PUSH X	: CALL PRINT MESSAGE

0871 D9		EXX	:
0872 213F07		LXI H,073FH	:
0875 E5		PUSH H	:
0876 2500		MVI L,0	:
0878 26FF		MVI H,0FFH	:
087A E5		PUSH H	:
087B 210000		LXI H,0	:
087E E5		PUSH H	:
087F E5		PUSH H	:
0880 D9		EXX	:
0881 CD990F		CALL LEAVER	:
0884 DDE1		POP X	:
0886 FDC802EE		SET 5,2(Y)	: SIGNAL CLEAR SCREEN
088A CDAA08		CALL WAITK	: CALL 08AA
088D DDE5		PUSH X	: SAVE HEADER POINTER
088E 111100		LXI D,11H	: SIZE
0892 AF		XRA A	: HEADER FLAG
0893 CD6800		CALL SAVBYT	:
0894 DDE1		POP X	: NOW DELAY FOR 50 INTS
0898 0632		MVI B,32H	:
089A 76	LOOP00	HLT	: WAIT FOR IRQ
089B 10FD		DJNZ LOOP00	:
089D DD5E08		MOV E,11(X)	: GET THE LENGTH
08A0 DD560C		MOV D,12(X)	:
08A3 3EFF		MVI A,0FFH	: SET WRITE DATA BLOCK
08A5 DDE1		POP X	: GET START
08A7 C36800		JMP SAVBYT	: AND WRITE IT
08AA F5	WAITK	PUSH PSW	
08AB C5		PUSH B	
08AC D5		PUSH D	
08AD 01409C		LXI B,9C40H	
08B0 0B	LOOP01	DCX B	
08B1 79		MOV A,C	: BC=0?
08B2 B0		ORA B	
08B3 20FB		JRNZ LOOP01	
08B5 AF	LOOP02	XRA A	
08B6 DBFE		IN 0FEH	
08B8 E61F		ANI 01FH	
08BA FE1F		CFI 01FH	
08BC 28F7		JRZ LOOP02	: F7
08BE DDE5		PUSH X	
08C0 D9		EXX	
08C1 21A90B		LXI H,08A9H	
08C4 E5		PUSH H	
08C5 2E00		MVI L,0	
08C7 26FF		MVI H,0FFH	
08C9 E5		PUSH H	
08CA 210000		LXI H,0	
08CD E5		PUSH H	
08CE E5		PUSH H	

```

08CE D9          EXX
08D0 CD990F      CALL LEAVER

08D3 DDE1        POP X
08D5 D1          POP D
08D6 C1          POP B
08D7 F1          POP PSW
08D8 C9          RET

08D9 D9          REPORTC  EXX          ;CALLS 1BED IN HOME ROM
08DA 21ED1B      LXI H,1BEDH          ;WHICH RESULTS IN AN ERROR 12
08DD E5          PUSH H
08DE 2EQ0        MVI L,0
08E0 26FF        MVI H,0FFH
08E2 E5          PUSH H
08E3 D9          EXX
08E4 CD8A0F      CALL OFBAH

```

```

08E7 21EA5E      INITI    LXI H,5EEAH          ;SYSTEM INITIALIZE
08EA 22BC5C      SHLD SYSCON
08ED CDC41B      CALL FIXTOP

```

```

0F99            ORG 0F99H
0F99 F5          LEAVER   PUSH PSW          ;EXIT TO HOME ROM
0F9A 3AC25C      LDA VIDMOD          ;5CC2
0F9D A7          ANA A
0F9E 2804        JRZ LEAV01
0FA0 F1          POP PSW
0FA1 C390FD      JMP OFD90H
0FA4 F1          LEAV01   POP PSW
0FA5 C3D065      JMP 65D0H

```

```

;*****
; **
; **      JUMP TABLE FOR DISPATCHER
; **
;*****

```

```

1FE0            ORG 1FE0H
1FE0 5E64        DW 645EH          ;RAM SERVICE
1FE2 0564        DW 6405H          ; "
1FE4 FFFF        DW 0FFFFH        ;RESERVED
1FE6 FFFF        DW 0FFFFH        ;RESERVED
1FE8 FFFF        DW 0FFFFH        ;RESERVED
1FEA FFFF        DW 0FFFFH        ;RESERVED
1FEC E500        DW SALDRTN        ;SAVE/LOAD RETURN
1FEE A30E        DW 0EA3H          ; ??

```

1FF0 5108	DW SAVCNT	;SAVE CONTROL
1FF2 E506	DW MECNT	;MERGE CONTROL
1FF4 CC05	DW LDCNT	;LOAD CONTROL
1FF6 AB01	DW SAVEETC	;SORT OPERATION FOR TYPE
1FF8 8D01	DW LDEDG1	;GET TAPE EDGE
1FFA 8901	DW LDEDG2	; " " "
1FFC EC00	DW LD_BYTES	;LOAD DATA FROM TAPE
1FFE 6800	DW SAVBYT	;SAVE DATA TO TAPE

INCLUDE MID.ASM

0000	HEDMRK	EQU 0	;FIRST BYTE IN DATA BLOCK
2000			;FLAG FOR HEADER
0001	DATMRK	EQU 1	; " DATA BLOCK
0002	EOFMRK	EQU 2	;POSITION BLOCK
0016	SYNCD	EQU 16H	;FOR BYTE SYNC
0040	ESFRED	EQU 40H	
0040	ESFWRT	EQU ESFRED	
0001	MOTOR	EQU 1	
0002	WRITEN	EQU 2	
0080	WRTDAT	EQU 80H	
5E80	VERFLG	EQU 5E80H	;NZ=VERIFY FLAG
5E81	FILPOS	EQU VERFLG+1	
FFFF	ATEMP	EQU 0FFFFH	
5E82	DTEMP	EQU FILPOS+1	
5E84	XTEMP	EQU DTEMP+2	
5E86	START	EQU XTEMP+2	
5E88	INPROG	EQU START+2	
5E89	DRVNUM	EQU INPROG+1	

1630 ORG 1630H

```

;*****
;**
;** JUMP TABLE FOR TAPE OPERATION
;**
;*****

```

1630 01 DB 1 ;RELEASE NUMBER

1631 C36816	JMP ESFHEDR
1634 C3AE16	JMP ESFLOAD
1637 C38917	JMP RPREAM
163A C3F616	JMP REDBLK
163D C3EB17	JMP REDBYTE
1640 C3F918	JMP FINDPOS
1643 C35A18	JMP ESFSAVE
1646 C31919	JMP WRITDATA
1649 C32F19	JMP WPREAM
164C C37019	JMP WBLOCK
164F C3051A	JMP WRBYT
1652 C3791A	JMP WRBITS

```

1655 C3F31A      JMP FORMAT
1658 C3B11A      JMP ALLOFFE

```

```

;*****
;**
;** CHECK FOR LOAD HEADER FROM SF OR CASS
;**
;*****

```

```

165B      LODHDR   EQU $           ;X=HEADER
165B F5      PUSH PSW
165C DD7EFO   MOV A,-16(X)         ;GET FIRST CHAR OF NAME DESIRED
165F FE40     CPI '0'
1661 2805     JRZ ESFHEDR          ;IF ESF GET THE HEADER
1663 F1       POP PSW
1664 C3FC00   JMP LD BYTES         ;ELSE GET FROM CASSETTE
1667 17       DB 23

```

```

;*****
;**
;** LOAD HEADER FROM STRINGY
;** ENTER X=WHERE TO PUT IT
;** DE=LEN ALWAYS 17 INCLUDING THE TYPE
;** IF AN ERROR OCCURS JUMP TO REPORTR (TWICE AROUND NO GOT)
;** IF BREAK THEN JUMP TO REPORT D
;** ELSE RETURN W/CARRY SET AND
;**
;*****

```

```

1668      ESFHEDR  EQU $
1668 0E40     MVI C,ESFRED         ;PORT ADDRESS
166A 3A895E   LDA DRVNUM
166D A7       ANA A
166E 2801     JRZ NOTONE
1670 0C       INR C
1671      NOTONE  EQU $
1671 F3       DI
1672 AF       XRA A
1673 32805E   STA VERFLG
1676 F1       POP PSW
1677 DDE5     PUSH X
1679      ESFH00  EQU $
1679 CD8917   CALL RPREAM         ;GET A PREAMBLE
167C 7A       MOV A,D             ;CHECK FOR HEADER
167D DE00     SBI HEDMRK
167F C27916   JNZ ESFH00
1682 211100   LXI H,11H          ;LENGTH
1685 3E03     MVI A,3
1687 3D       BLOOP  DCR A        ;16A+55
1688 20FD     JRNZ BLOOP
168A 3E7F     MVI A,07FH         ;CARRY CLEAR IF BREAK

```

```

168C DBFF          IN OFFH
168E 2F           CMA
168F 1F           RAR
1690 300A         JRNC BAEND          ; IF BREAK THEN SPLIT
1692 CDB11A       CALL ALLOFFE
1695 C3F800       JMP REPORTD
1698             BAEND EQU $
1698 CDF616       CALL REDBLK
169B DDE1         POP X
169D 7C           MOV A,H
169E B5           ORA L
169F C2E21A       JNZ REDERR
16A2 CDBA1A       CALL ALLOFF
16A5 3EFF         MVI A,OFFH
16A7 32885E       STA INPROG
16AA AF          XRA A
16AB 37           STC
16AC FB          EI
16AD C9          RET

```

```

; *****
; **
; ** LOAD OR VERIFY DATA BLOCK (CARRY IF LOAD)
; ** ENTER X=START DE=LENGTH
; ** THIS IS CALLED AFTER A SUCCESSFUL COMPARE OF HEADER
; ** AND THE MOTOR WILL BE ON
; ** IF PARITY OR CHECKSUM ERROR THEN REPORT R
; ** BREAK = REPORT D
; ** SUCCESSFUL SET Z AND C
; **
; *****

```

```

16AE             ESFLOAD EQU $          ; TEMP
16AE F5          PUSH PSW
16AF 3A895E       LDA INPROG
16B2 A7          ANA A
16B3 C2BA16       JNZ GOTESF
16B6 F1          POP PSW
16B7 C3FC00       JMP LD BYTES
16BA 0E40         GOTESF MVI C,ESFRED
16BC 3A895E       LDA DRVNUM
16BE A7          ANA A
16C0 2801         JRZ NUTONE
16C2 0C          INR C
16C3 F1          POP PSW
16C4 F3          DI
16C5 3E00         MVI A,0
16C7 3801         JRC NOTVER
16C9 3C          INR A
16CA 32805E       NOTVER STA VERFLG

```

TIMEX 2068 STRINGY FLOPPY

PAGE 33

COPYRIGHT (C) 1984 JIM MAYNARD

```

16CD 00      ESFL00  NOP
16CE D5      PUSH D
16CF CD8917   CALL RFREAM
16D2 7A      MOV A,D
16D3 DE01     SBI DATMRK
16D5 D1      POP D
16D6 00      NOP ;X= ADDRESS
16D7 00      NOP
16D8 00      NOP
16D9 C2CD16   JNZ ESFL00
16DC D5      PUSH D
16DD E1      POP H
16DE DD23     INX X
16E0 DD2B     DCX X
16E2 DD23     INX X
16E4 DD2B     DCX X
16E6 3E09     MVI A,9
16E8 CDF616   CALL REDBLK
16EB CDB11A   CALL ALLOFF
16EE 7C      MOV A,H
16EF B5      ORA L
16F0 C2E21A   JNZ REDERR
16F3 AF      XRA A
16F4 37      STC
16F5 C9      RET

```

; ** READ OR VERIFY DATA BLOCK

; ** X=START

; ** H=LEN

; ** VERDAT=0 LOAD

```

16F6      REDBLK  EQU $
16F6 00      NOP
16F7 00      NOP
16F8 00      NOP
16F9 00      NOP
16FA ED78     INP A
16FC FA0417   JM QB0
16FE 3EFF     MVI A,0FFH
1701 B7      ORA A
1702 184B     JMPR QUEND
1704 ED58     QB0  INP E ;CATCH BIT ONE START
1706 FA0417   JM QB0
1709 3AFFFF   LDA ATEMP ;ADD CHECKSUM TO PREV.BYTE
170C B2      ADD D
170D 32FFFF   QB1  STA ATEMP
1710 0608     MVI B,8 ;BIT COUNT
1712 0608     MVI B,8
1714 3E05     QB2  MVI A,5
1716 C31917   JMP QB3

```

1719	QB3	EQU \$	:
1719 00		NOP	
171A 00		NOP	
171B 00		NOP	
171C 00		NOP	
171D 3D	ALOOP	DCR A	;16A+55
171E 20FD		JRNZ ALOOP	
1720 3E7F		MVI A,07FH	;CARRY CLEAR IF BREAK
1722 DBFE		IN OFEH	
1724 2F		CMA	
1725 1F		RAR	
1726 3006		JRNC SAEND	;IF BREAK THEN SPLIT
1728 CDB11A		CALL ALLOFFE	
172B C3F800		JMP REPORTD	
172E	SAEND	EQU \$	
172E 7B		MOV A,E	:
172F 2F		CMA	:
1730 ED58		INP E	;THE DECISION AFTER
1732 FA3D17		JM QB5	;THIS DOESNT COUNT
1733 ED58	QB4	INP E	
1737 F23517		JP QB4	
173A C34517		JMP QB6	
173D ED58	QB5	INP E	:
173F FA3D17		JM QB5	
1742 C34517		JMP QB6	
1745 AB	QB6	XRA E	
1746 07		RLC	
1747 CB1A		RARR D	
1749 AF		XRA A	
174A 10CB		DJNZ QB2	
174C 00		NOP	
174D 00		NOP	
174E 00		NOP	
174F	QUEND	EQU \$	
174F C0		RNZ	;RETURN IF PARITY
1750 3A805E		LDA VERFLG	
1753 B7		ORA A	
1754 7A		MOV A,D	
1755 CA5F17		JZ REDBLO	;IF A LOAD
1758 DD8E00		CMP 0(X)	
175B C0		RNZ	;ERROR
175C C36617		JMP REDBL1	
175F DD7700	REDBLO	MOV 0(X),A	
1762 3E00		MVI A,0	
1764 3E00		MVI A,0	
1766 DD23	REDBL1	INX X	;ADJUST COUNT
1768 2B		DCX H	
1769 DD23		INX X	;DELAY
176B DD2B		DCX X	
176D DD7E00		MOV A,0(X)	
1770 7C		MOV A,H	;CHECK FOR ZERO

```

1771 B5          ORA L
1772 C2F616      JNZ REDBLK
1775 CDEB17      CALL REDBYTE          ;GET CHECKSUM
1778 C0          RNZ
1779 3E04        MVI A,4
177B CD3D18      CALL WAIT
177E 00          NOP
177F 00          NOP
1780 CDEB17      CALL REDBYTE          ;ADD BYTE IN D TO
1783 C0          RNZ                  ;PREVIOUS TOTAL
1784 3AFFFF      LDA ATEMP            ;CHECKSUM+/CHECKSUM=0
1787 B7          ORA A
1788 C9          RET

```

```

;*****
; **
; **   READ PREAMBLE, SYNC, TYPE
; **
;*****

```

```

1789          RPREAM EQU $          ;ENTER
1789 3E01      MVI A,MOTOR
178B ED79      OUTP A
178D CDBF1A    CALL MOTDELAR
1790 CDA617    RPROO CALL PR1
1793 7A        MOV A,D
1794 DE16      SBI SYNCOD          ;REM 140 BETWEEN CALLS
1796 C29017    JNZ RPROO          ;
1799 3E04      MVI A,4
179B CD3D18    CALL WAIT          ;
179E 00        NOP
179F CDEB17    CALL REDBYTE        ;
17A2 32FFFF    STA ATEMP          ;CHECKSUM
17A5 C9        RET                ;
17A6 0614      PR1 MVI B,20        ;TRY FOR 20 0 BEFORE ONE
17A8 CD4018    PR2 CALL CHKBK
17AB ED78      INP A
17AD F2A817    JF PR2
17B0 CD4018    PR3 CALL CHKBK
17B3 ED78      INP A
17B5 FAB017    JM PR3              ;+
17B8 ED78      PR4 INP A
17BA FAA617    JM PR1
17BD ED78      PR5 INP A          ;
17BF F2BD17    JP PR5            ;

```

17C3 3E09		MVI A,9	:
17C4 CD3D18		CALL WAIT	; (16A)+55=199
17C7 ED78		INP A	:
17C9 F2A617		JP PR1	; SHOULD BE A ZERO
17CC ED78	PR6	INP A	:
17CE FACC17		JM PR6	:
17D1 C3D417		JMP PR7	:
17D4 3E07	PR7	MVI A,7	:
17D6 CD3D18		CALL WAIT	:
17D9 10DD		DJNZ PR4	; 13 UP 8 DOWN
17DB 04		INR B	; IF CHANGE TO - THEN ONE
17DC ED78		INP A	:
17DE F2BD17		JP PR5	; ELSE TRY AGAIN
17E1 ED58	PR8	INP E	:
17E3 FAE117		JM PR8	:
17E6 3E00		MVI A,0	:
17E8 C3FD17		JMP RB1	; AND GET THE SYNC CODE
17EB	REDBYTE	EQU \$	; CHECK PARITY OF PREV. BYTE
17EB ED78		INP A	:
17ED FAF417		JM RBO	:
17F0 3EFF		MVI A,0FFH	:
17F2 B7		ORA A	:
17F3 C9		RET	:
17F4 ED58	RBO	INP E	; CATCH BIT ONE START
17F6 FAF417		JM RBO	:
17F9 3AFFFF		LDA ATEMP	; ADD CHECKSUM TO PREV.BYTE
17FC 82		ADD D	:
17FD 32FFFF	RB1	STA ATEMP	:
1800 0608		MVI B,8	; BIT COUNT
1802 0608		MVI B,8	:
1804 3E05	RB2	MVI A,5	:
1806 C30918		JMP RB3	:
1809	RB3	EQU \$	:
1809 00		NOP	:
180A 00		NOP	:
180B 00		NOP	:
180C 00		NOP	:
180D 3D	CLOOP	DCR A	; 16A+55
180E 20FD		JRNZ CLOOP	:
1810 3E7F		MVI A,07FH	; CARRY CLEAR IF BREAK
1812 DBFE		IN OFEH	:
1814 2F		CMA	:
1815 1F		RAR	:
1816 3006		JRNC CAEND	; IF BREAK THEN SPLIT
1818 CDB11A		CALL ALLOFFE	:
181B C3FB00		JMP REPORTD	:
181E	CAEND	EQU \$	:
181E 7B		MOV A,E	:
181F 2F		CMA	:
1820 ED58		INP E	; THE DECISION AFTER

1822 FA2D18 JM RB5 ;THIS DOESNT COUNT

1825 ED58 RB4 INP E

1827 F22518 JP RB4

182A C33518 JMP RB4

182D ED58 RB5 INP E ;

182F FA2D18 JM RB5

1832 C33518 JMP RB4

1835 AB RB6 XRA E

1836 07 RLC

1837 CB1A RARR D

1839 AF XRA A

183A 10CB DJNZ RB2

183C C9 RET

183D 3D WAIT DCR A ;16A+55

183E 20FD JRNZ WAIT

1840 3E7E CHKBK MVI A,07EH ;CARRY CLEAR IF BREAK

1842 DBFE IN OFEH

1844 2F CMA

1845 1E RAR

1846 D0 RNC ; IF BREAK THEN SPLIT

1847 CDB11A CALL ALLOFFE

184A C3F800 JMP REPORTD

;**

;** SAVE DATA BLOCK AND HEADER TO TAPE

;** ENTER X=HEADER LOCATION DE=LEN

;** AFTER WRITING HEADER THEN POP RETURN

;** E,11(X) AND D,12(X) DE=LEN

;** POP X=START OF DATA BLOCK

;** IF WRITE PROTECT THEN REPORT ?

;** IF END OF TAPE THEN REPORT ?

;** RET WITH Z=0

;**

184D GETMSG EQU \$;MESSAGE FOR SAVE ON TAPE

184D F5 PUSH PSW

184E 11893C LXI D,3C89H

1851 DD7E01 MOV A,1(X)

1854 FE40 CPI '0'

1856 2802 JRZ ESFSAVE

1858 F1 GETMXX POP PSW

1859 C9 RET

185A ESFSAVE EQU \$

TIMEX 2048 STRINGY FLOPPY

PAGE 38

COPYRIGHT (C) 1984 JIM MAYNARD

185A F1		POP PSW	; CLEAN STACK
185B E1		POP H	; OLD RETURN ADDRESS
185C E1		POP H	; START OF DATA BLOCK
185D F3		DI	
185E 0E40		MVI C,ESFRED	
1860 3A895E		LDA DRVNUM	
1863 A7		ANA A	
1864 2801		JRZ SKONE	
1866 0C		INR C	
1867 22845E	SKONE	SHLD START	
186A DD7E03		MOV A,3(X)	; CHECK COMMA
186D FE2C		CPI ','	
186E C2EE1A		JNZ BADFILE	
1872 ED78		INP A	; CHECK WRITE PROTECT
1874 E601		ANI 1	
1876 C2DD1A		JNZ WRTPRT	
1879 DD7E02	ESFS00	MOV A,2(X)	; GET POSITION
187C DE30		SBI 30H	
187E 32815E		STA FILPOS	
1881 DD7E04		MOV A,4(X)	
1884 FE2B		CPI '+'	
1886 2003		JRNZ ESFNF	; FORMAT
1888 C3F31A		JMP FORMAT	
188B 0607	ESFNF	MVI B,7	
188D DD5E0C		MOV E,12(X)	; CHECK FOR ZERO LEN
1890 DD7E0B		MOV A,11(X)	
1893 B3		ORA E	
1894 2004		JRNZ BEAST	
1896 AF		XRA A	
1897 37		STC	
1898 FB		EI	
1899 C9		RET	
189A DDE5	BEAST	PUSH X	
189C DD7E04	LOOPBS	MOV A,4(X)	
189F DD7702		MOV 2(X),A	
18A2 DD23		INX X	
18A4 10E6		DJNZ LOOPBS	
18A6 DD360220		MVI 2(X),20H	; BLANK THE LAST THREE
18AA DD360320		MVI 3(X),20H	
18AE CDF918		CALL FINDEOS	; FIND THE ONE BEFORE
18B1 DDE1		POP X	
18B3 DDE5		PUSH X	
18B5 2E00		MVI L,HEDMEK	; TYPE

TIMEX 2068 STRINGY FLOPPY
COPYRIGHT (C) 1984 JIM MAYNARD

PAGE 39

```

18CC DD560C      MOV D,12(X)
18CF DD2AB65E    LIXD START
18D3 2E01        MVI L,DATMRK
18D5 CD1919      CALL WRITDATA
18D8 C2EE18      JNZ ENDOFW
18DB 3E02        MVI A,WRITEN
18DD ED79        OUTP A
18DF CDC51A      CALL MOTDELAU
18E2 DD21815E    LXI X,FILPOS
18E4 110100      LXI D,1
18E9 2E02        MVI L,EOFMRK
18EB CD1919      CALL WRITDATA
18EE F5          ENDOFW    PUSH PSW
18EF CDBA1A      CALL ALLOFF
18F2 F1          POP PSW
18F3 37          STC
18F4 FB          EI
18F5 C8          RZ
18F6 C3DB1A      JMP ENDDET

```

*** POSITION TAPE FOR WRITE

```

18F9          FINDPOS EQU $          ;FIND EOF OF THE
18F9 CD8917    CALL RPREAM          ;ONE IN FILPOS
18FC C2F918    JNZ FINDPOS          ;EOT THEN LOOP
18FF 7A        MOV A,D
1900 DE02      SBI EOFMRK          ;IF NOT END OF FRAME
1902 C2F918    JNZ FINDPOS
1905 CDEB17    CALL REDBYTE          ;ELSE GET THE NEXT BYTE
1908 C2F918    JNZ FINDPOS
190B 21815E    LXI H,FILPOS          ;POINT AT DESIRED
190E 7A        MOV A,D
190F 3C        INR A                ;WE WANT THE NEXT ONW
1910 BE        CMP M
1911 C2F918    JNZ FINDPOS
1914 CDBA1A    CALL ALLOFF
1917 AF        XRA A
1918 C9        RET

```

```

=====
; ** WRITE PREAMBLE, SYNC, BLOCK TYPE, DATA, CHECKSUM
; ** AFTER TAPE HAS BEEN POSITIONED
; ** ENTER WITH "L"= TYPE
; ** X= START OF DATA
; ** DE= LENGTH
; ** RETURN NZ IF END OF TAPE
; ** AND MOTOR STOPPED

```

TIMEX 2068 STRINGY FLOPPY
 COPYRIGHT (C) 1984 JIM MAYNARD

PAGE 40

```

1 1919          WRITDATA EQU $          ; ENTER -----
2 1919 D5          PUSH D
3 191A CD2E19      CALL WPREAM          ; WRITE PREAMBLE
4 191D E1          POP H
5 191E 2006        JRNZ WRITEND
6 1920 CDBF1B      CALL DEL43
7 1923 CD7019      CALL WBLOCK
8 1926 F5          WRITEND PUSH PSW      ; SAVE CC'S
9 1927 AF          XRA A
10 1928 ED79        OUTP A              ; MOTOR OFF
11 192A CDC51A      CALL MOTDELAW
12 192D F1          POP PSW
13 192E C9          RET
14
15 ; =====
16
17 ; ** WRITE PREAMBLE SYNC AND BYTE IN L
18 ; ** EXITS THRU WRIBYT
19 ; ** RETURNS NZ IF EOT
20
21 192F 3E83        WPREAM MVI A,MOTOR+WRITEN+WRITDAT
22 1931 32FFFF      STA ATEMP
23 1934 ED79        OUTP A
24 1936 CDC51A      CALL MOTDELAW
25 1939 C0          RNZ
26 193A 1600        MVI D,0
27 193C 42          MOV B,D
28
29 193D CD791A      CALL WRBITS          ; WRITE 256 "0"'S
30 1940 C0          RNZ
31 1941 CDBF1B      CALL DEL47
32 1944 CD791A      CALL WRBITS          ; WRITE 256 "0"'S
33 1947 C0          RNZ
34 1948 CDBF1B      CALL DEL43
35 194B CDBF1B      CALL DEL47
36 194E 3AFFFF      LDA ATEMP
37 1951 E67E        ANI 7EH            ; WRITE A ONE AS THE PARITY
38 1953 ED79        OUTP A            ; BIT
39 1955 CDBC1B      CALL DEL55
40 1958 CDBD1B      CALL DEL51
41 195B 1616        MVI D,SYNCD        ; NOW THE SYNC MARK
42 195D CD051A      CALL WRBYT
43 1960 C0          RNZ
44 1961 CDBF1B      CALL DEL47
45 1964 CDBF1B      CALL DEL43
46 1967 55          MOV D,L            ; AND THE BYTE IN "L"
47 1968 1E00        MVI E,0            ; CLEAR CHECKSUM
48 196A C3051A      JMP WRBYT          ; AND SPLIT
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76

```

TIMEX 2068 STRINGY FLOPPY
COPYRIGHT (C) 1984 JIM MAYNARD

PAGE 41

```

; ** WRITE BLOCK AT (X)
; ** HL=COUNT
; ** THEN CHECKSUM 3 TIMES
; ** EXITS BY FALLING THRU WRBYT
; ** RETURNS NZ IF END OF TAPE

```

```

196D      WRLOOP      EQU $
196D CDC01B      CALL DEL39

```

```

1970      WBLOCK      EQU $      ; ENTER -----
1970 DD5600      MOV D,0(X)      ; GET THE BYTE
1973 3AFFFF      LDA ATEMP      ; SEND IT
1976 00          NOP

```

```

1977 3AFFFF      LDA ATEMP      ; WRITE FIRST CLOCK AFTER PARITY
197A ED79      OUTP A

```

```

197C 7A          MOV A,D          ; ADD IN CHECKSUM
197D 83          ADD E
197E 5E          MOV E,A

```

```

197F 0604      MVI B,4          ; DELAY
1981 10FE      DJNZ $
1983 00          NOP

```

```

1984 00          NOP
1985 0608      MVI B,8          ; BIT COUNT
1987 00          NOP

```

```

1988 00          NOP
1989 180A      JMPR QRBITS      ; WRITE THE BYTE
198B          QRBITS      EQU $      ; DELAY

```

```

198B 3E03      MVI A,3
198D 3D          DCR A
198E 20ED      JRNZ QOPO

```

```

1990 ED78      INP A
1992 E602      ANI 2
1994 C0          RNZ

```

```

1995 3AFFFF      QRBITS      LDA ATEMP      ; ENTER HERE -----
1998 CB0A      RRCR D
199A 00          NOP

```

```

199B 00          NOP
199C 302C      JRNC QRTZER
199E EE80      XRI 80H

```

```

19A0 ED79      OUTP A          ; FOR A ONE BIT
19A2 EE80      QRCCLK      XRI 80H      ; TOGGLE WRITE DATA
19A4 32FFFF      STA ATEMP

```

```

19A7 00          NOP
19AB 00          NOP
19AD 00          NOP
19AA DD23      INX X
19AC DD2B      DCX X

```

TIMEX 2068 STRINGY FLOPPY
COPYRIGHT (C) 1984 JIM MAYNARD

PAGE 42

19AE DD23	INX X	
19B0 DD2B	DCX X	
19B2 DD23	INX X	
19B4 DD2B	DCX X	
19B6 DD23	INX X	
19B8 DD2B	DCX X	
19BA DD23	INX X	
19BC DD2B	DCX X	
19BE 00	NOP	
19BF 00	NOP	
19C0 BF	CMP A	
19C1 ED79	OUTP A	; CLOCK FOR NEXT BIT
19C3 00	NOP	
19C4 00	NOP	
19C5 00	NOP	
19C6 10C3	DJNZ GRBITL	
19C8 1804	JMPR QN00	; EXIT -----
19CA 00	QRTZER	NOP
19CB C3A219	JMP DRCLK	
19CE C0	QN00	RNZ ; RETURN IF END OF TAPE
19CF 0605	MVI B,5	; NOW THE MIDDLE
19D1 10FE	DJNZ \$	; OF THE PARITY BIT
19D3 3AFFFF	LDA ATEMP	; AND THE LEVEL OF THE NEXT
19D6 F680	ORI 80H	; CLOCK WILL BE HIGH
19D8 32FFFF	STA ATEMP	
19DB E67F	ANI 7FH	
19DD ED79	OUTP A	
19DF AF	XRA A	; SET OK CODE AND RETURN
19E0 00	NOP	; WITH 14 CYCLES
19E1 00	NOP	
19E2 00	NOP	
19E3 00	NOP	; IF EOT
19E4 DD23	INX X	; NEXT
19E6 2B	DCX H	; COUNTER-1
19E7 7C	MOV A,H	
19E8 B5	DRA L	
19E9 2082	JRNZ WRLOOP	; TILL BLOCK DONE
19EB CDBD1B	CALL DEL51	; NOW WRITE CHECKSUM (NEG)
19EE 97	SUB A	
19EF 93	SUB E	
19F0 57	MOV D,A	
19F1 CD051A	CALL WRBYT	; ONCE
19F4 C0	RNZ	
19F5 CDBD1B	CALL DEL51	
19F8 CDBF1B	CALL DEL43	
19FB CD051A	CALL WRBYT	; TWICE
19FE C0	RNZ	
19FF CDBC1B	CALL DEL55	
1A02 CDBC1B	CALL DEL55	
1A05		; AND FALL THRU FOR THREE

TIMEX 2068 STRINGY FLOPPY
 COPYRIGHT (C) 1984 JIM MAYNARD

PAGE 43

=====

;** WRITE BYTE IN "D"
 ;** ADD CHECKSUM AND WRITE PARITY BIT

1A05	WRBYT	EQU \$	;ENTER-----
1A05 3AFFFE		LDA ATEMP	;WRITE FIRST CLOCK AFTER PARITY
1A08 ED79		OUTP A	
1A0A 7A		MOV A,D	;ADD IN CHECKSUM
1A0B 83		ADD E	
1A0C 5F		MOV E,A	
1A0D 0604		MVI B,4	;DELAY
1A0E 10FE		DJNZ \$	
1A11 00		NOP	
1A12 00		NOP	
1A13 0608		MVI B,8	;BIT COUNT
1A15 00		NOP	
1A16 00		NOP	
1A17 180A		JMPR SRBITS	;WRITE THE BYTE
1A19	SRBITL	EQU \$	;DELAY
1A19 3E03		MVI A,3	
1A1B 3D	SOPO	DCR A	
1A1C 20FD		JRNZ SOPO	
1A1E ED78		INP A	
1A20 E602		ANI 2	
1A22 C0		RNZ	
1A23 3AFFFE	SRBITS	LDA ATEMP	;ENTER HERE -----
1A26 C80A		RRCR D	
1A28 00		NOP	
1A29 00		NOP	
1A2A 302C		JRNC SRTZER	
1A2C EE90		XRI 80H	
1A2E ED79		OUTP A	;FOR A ONE BIT
1A30 EE80	SRCLK	XRI 80H	;TOGGLE WRITE DATA
1A32 32FFFF		STA ATEMP	
1A35 00		NOP	
1A36 00		NOP	
1A37 00		NOP	
1A38 DD23		INX X	
1A3A DD2B		DCX X	
1A3C DD23		INX X	
1A3E DD2B		DCX X	
1A40 DD23		INX X	
1A42 DD2B		DCX X	
1A44 DD23		INX X	
1A46 DD2B		DCX X	
1A48 DD23		INX X	
1A4A DD2B		DCX X	

TIMEX 2068 STRINGY FLOPPY
COPYRIGHT (C) 1984 JIM MAYNARD

1A4C 00		NOP	
1A4D 00		NOP	
1A4E BF		CMP A	
1A4F ED79		OUTP A	;CLOCK FOR NEXT BIT
1A51 00		NOP	
1A52 00		NOP	
1A53 00		NOP	
1A54 10C3		DJNZ SRBITL	
1A56 1804		JMPR SNOO	;EXIT
1A58 00	SRTZER	NOP	
1A59 C3301A		JMP SRCLK	
1A5C C0	SNOO	RNZ	;RETURN IF END OF TAPE
1A5D 0605		MVI B,5	;NOW THE MIDDLE
1A5F 10FE		DJNZ \$	;OF THE PARITY BIT
1A61 3AFFFF		LDA ATEMP	; AND THE LEVEL OF THE NEXT
1A64 F680		ORI 80H	; CLOCK WILL BE HIGH
1A66 32FFFF		STA ATEMP	
1A69 E67F		ANI 7FH	
1A6B ED79		OUTP A	
1A6D AF		XRA A	;SET OK CODE AND RETURN
1A6E C9		RET	;WITH 14 CYCLES

```

;=====
; ** WRITE "B" BITS IN "D"
; ** C= OUTPUT PORT
; ** CALLING PROG HAS 52 TS BETWEEN CALLS TO DO SOME THING
; ** FROM CALL TO TRANSITION OF ONE IS 75 TS OF 155
; ** FROM CLOCK OF LAST + RETURN IS 28 TS
; ** RETURNS NZ IF END OF TAPE

```

1A6F	WRBITL	EQU \$	;DELAY
1A6F 3E03		MVI A,3	
1A71 3D	DOPO	DCR A	
1A72 20FD		JRNZ DOPO	
1A74 ED78		INP A	
1A76 E602		ANI 2	
1A78 C0		RNZ	
1A79 3AFFFF	WRBITS	LDA ATEMP	;ENTER HERE
1A7C CBOA		RRCR D	
1A7E 00		NOP	
1A7F 00		NOP	
1A80 302B		JRNC WRTZER	
1A82 EE80		XRI 80H	
1A84 ED79		OUTP A	;FOR A ONE BIT
1A86 EE80	WRCLK	XRI 80H	;TOGGLE WRITE DATA
1A88 32FFFF		STA ATEMP	
1A8B DD23		INX X	
1A8D DD2B		DCX X	

TIMEX 2068 STRINGY FLOPPY
COPYRIGHT (C) 1984 JIM MAYNARD

PAGE 45

1A8F DD23	INX X	
1A91 DD2B	DCX X	
1A93 DD23	INX X	
1A95 DD2B	DCX X	
1A97 DD23	INX X	
1A99 DD2B	DCX X	
1A9B DD23	INX X	
1A9D DD2B	DCX X	
1A9F 00	NOP	
1AA0 00	NOP	
1AA1 00	NOP	
1AA2 00	NOP	
1AA3 00	NOP	
1AA4 BF	CMP A	
1AA5 ED79	OUTP A	;CLOCK FOR NEXT BIT
1AA7 00	NOP	
1AA8 00	NOP	
1AA9 00	NOP	
1AAA 10C3	DJNZ WRRITL	
1AAC C9	RET	;EXIT -----
1AAD 00	WRTZER NOP	
1AAE C38A1A	JMP WRCLK	

```

;*****
; **
; **  MOTOR ON ETC.
; **
;*****

```

1AB1 CDBA1A	ALLOFF	CALL ALLOFF
1AB4 32885E		STA INPROG
1AB7 37		STC
1AB8 FB		EI
1AB9 C9		RET
1ABA 3E00	ALLOFF	MVI A,0
1ABC ED79		OUTP A
1ABE C9		RET
1ABF	MOTDELA	EQU \$
1ABF E5		PUSH H
1AC0 212823		LXI H,9000
1AC3 1804		JMPR MOTLOOP
1AC5	MOTDELA	EQU \$
1AC5 E5		PUSH H
1AC6 21C832		LXI H,13000
1AC9 ED78	MOTLOOP	INP A
1ACB E602		ANI 2

TIMEX 2068 STRINGY FLOPPY
COPYRIGHT (C) 1984 JIM MAYNARD

```

1ADD C2D41A      JNZ MOTEND
1AD0 2B          DCX H
1AD1 7C          MOV A,H
1AD2 B5          ORA L
1AD3 C2C91A      JNZ MOTLOOP
1AD6 E1          MOTEND POP H
1AD7 C9          RET

```

```

1ADB CDB11A      ENDOFT CALL ALLOFFE
1ADB CF          RST 1
1ADC 07          DB 07H
1ADD CDB11A      WRTPRT CALL ALLOFFE
1AE0 CF          RST 1
1AE1 17          DB 17H
1AE2 CDB11A      REDERR CALL ALLOFFE
1AE5 CF          RST 1
1AE6 1A          DB 1AH
1AE7 CE          FILPOSR RST 1
1AE8 12          DB 12H
1AE9 CDB11A      BRKERR CALL ALLOFFE
1AEC CE          RST 1
1AED 0C          DB 0CH
1AEE CDB11A      BADFILE CALL ALLOFFE
1AF1 CE          RST 1
1AF2 0E          DB 0EH

```

```

;*****
; **
; **  FORMAT NEW TAPE
; **
;*****

```

```

1AF3          FORMAT EQU $ ;FORMAT NEW TAPE
1AF3 3E01      MVI A,MOTOR ;MOTOR ON
1AF5 ED79      OUTP A
1AF7 CD4018    EOTLOOP CALL CHKBK ;GO TO BEGINNING OF TAPE
1AFA ED78      INP A ;
1AFC E602      ANI 2 ;
1AFE 28F7      JRZ EOTLOOP ;
1B00 AF        XRA A
1B01 32B15E    STA FILPOS ;WRITE END OF FILE ZERO
1B04 3E83      MVI A,WRITEN+WRTDAT+MOTOR
1B06 ED79      OUTP A
1B08 32FFFF    STA ATEMP
1B0B CDE21B    CALL EOTDELY
1B0E 0605      MVI B,5
1B10 CDC51A    INDYGAP CALL MOTDELAU
1B13 10FB      DJNZ INDYGAP
1B15 DD21815E  WRTEOF LXI X,FILPOS
1B19 110100    LXI D,1

```

TIMEX 2068 STRINGY FLOPPY
 COPYRIGHT (C) 1984 JIM MAYNARD

PAGE 47

1B1C 2E02		MVI L,EOEMRK	
1B1E CD1919		CALL WRITDATA	
1B21 3E83		MVI A,WRITEN+WRTDAT+MOTOR	
1B23 ED79		OUTP A	
1B25 32FFFF		STA ATEMP	
1B28 16FF	ERASE6	MVI D,OFFH	
1B2A C32D1B		JMP ERASE0	
1B2D	ERASE0	EQU \$	
1B2D 3AFFFF		LDA ATEMP	;WRITE FIRST CLOCK AFTER PARITY
1B30 ED79		OUTP A	
1B32 7A		MOV A,D	;ADD IN CHECKSUM
1B33 83		ADD E	
1B34 5F		MOV E,A	
1B35 0604		MVI B,4	;DELAY
1B37 10FE		DJNZ \$	
1B39 00		NOP	
1B3A 00		NOP	
1B3B 0608		MVI B,8	;BIT COUNT
1B3D 180B		JMPR FRBITS	;WRITE THE BYTE
1B3F	FRBITL	EQU \$	;DELAY
1B3F 3E03		MVI A,3	
1B41 3D	EQEO	DCR A	
1B42 20FD		JRNZ FQPO	
1B44 ED78		INP A	
1B46 E602		ANI 2	
1B48 204E		JRNZ FNO1	
1B4A 3AFFFF	FRBITS	LDA ATEMP	;ENTER HERE
1B4D CB0A		RPCR D	
1B4F 00		NOP	
1B50 00		NOP	
1B51 00		NOP	
1B52 302A		JRNC FRTZER	
1B54 EE80		XRI 80H	
1B56 ED79		OUTP A	;FOR A ONE BIT
1B58 EE80	FRCLK	XRI 80H	;TOGGLE WRITE DATA
1B5A 32FFFF		STA ATEMP	
1B5D DD23		INX X	
1B5F DD2B		DCX X	
1B61 DD23		INX X	
1B63 DD2B		DCX X	
1B65 DD23		INX X	
1B67 DD2B		DCX X	
1B69 DD23		INX X	
1B6B DD2B		DCX X	
1B6D DD23		INX X	
1B6F DD2B		DCX X	
1B71 00		NOP	
1B72 00		NOP	
1B73 00		NOP	
1B74 BF		CMP A	
1B75 ED79		OUTP A	;CLOCK FOR NEXT BIT

TIMEX 2068 STRINGY FLOPPY
COPYRIGHT (C) 1984 JIM MAYNARD

1B77 00		NOP	
1B78 00		NOP	
1B79 00		NOP	
1B7A 10C3		DJNZ FRBITL	
1B7C 1804		JMPR FN00	;EXIT -----
1B7E 00	FRTZER	NOP	
1B7E C3581B		JMP FRCLK	
1B82 00	FN00	NOP	;RETURN IF END OF TAPE
1B83 0405		MVI B,5	;NOW THE MIDDLE
1B85 10FE		DJNZ \$	;OF THE PARITY BIT
1B87 00		NOP	;WILL BE SET LOW
1B88 3AFFFF		LDA ATEMP	; AND THE LEVEL OF THE NEXT
1B8B E680		DBI 80H	; CLOCK WILL BE HIGH
1B8D 32FFFF		STA ATEMP	
1B90 E67F		ANI 7FH	
1B92 ED79		OUTP A	
1B94 AF		XRA A	;SET OK CODE AND RETURN
1B95 C3981B		JMP FN01	;WITH 14 CYCLES
1B98 00	FN01	NOP	
1B99 DD23		INX X	
1B9B DD2B		DCX X	
1B9D DD23		INX X	
1B9F DD2B		DCX X	
1BA1 DD23		INX X	
1BA3 DD2B		DCX X	
1BA5 DD23		INX X	
1BA7 DD2B		DCX X	
1BA9 CA281B		JZ ERASEG	
1BAC CDB11A		CALL ALLOFFE	
1BAF AF		XRA A	
1BB0 37		STC	
1BB1 C9		RET	
1BB2	EOTDELY	EGU \$	
1BB2 3EFA		MVI A,250	
1BB4 10FE	EOTDFF	DJNZ \$	
1BB6 3D		DCR A	
1BB7 B7		ORA A	
1BB8 C2B41B		JNZ EOTDFF	
1BBB C9		RET	
; ** DELAYS			
1BBC 00	DEL55	NOP	
1BBD 00	DEL51	NOP	
1BBE 00	DEL47	NOP	
1BBF 00	DEL43	NOP	
1BC0 00	DEL39	NOP	
1BC1 00	DEL35	NOP	
1BC2 00	DEL31	NOP	

TIMEX 2068 STRINGY FLOPPY
COPYRIGHT (C) 1984 JIM MAYNARD

PAGE 49

1BC3 C9 DEL27 RET

1BC4 FIXTOP EQU *

1BC4 E5 PUSH H

1BC5 F5 PUSH PSW

1BC6 AF XRA A

1BC7 32895E STA DRVNUM

1BCA 32885E STA INPROG

1BCD E1 POP PSW

1BCE E1 POP H

1BCF C3F409 JMP 09F4H

1BD2 434F5059 DB 'COPYRIGHT (C) 1984 JIM MAYNARD'

END

0000 END

TIMEX 2068 STRINGY FLOPPY
---- SYMBOL TABLE ----

PAGE 50

ALLOFF	1ABA	ESFLOO	16CD	LD_LEADR	0126
ALLOFFE	1AB1	ESFLOAD	16AE	LD_LOOK	04D6
ALLOOP	171D	ESFNFF	188B	LD_LOOP	014F
ATEMP	FFFF	ESFRED	0040	LD_MARKR	016E
BADFILE	1AEE	ESFS00	1879	LD_NAME	053D
BAEND	1698	ESFSAVE	185A	LD_NEXT	0168
BEAST	189A	ESFWRT	0040	LD_PRG1	06D5
BLOOP	1687	E_LIN	5C59	LD_PROG	0673
BORDCR	5C48	E_PPC	5C49	LD_SAMP	0193
BREG	5C67	FILNERR	0228	LD_START	0112
BRKERR	1AE9	FILPOS	5E81	LD_SYNC	0135
CAEND	181E	FILPOSR	1AE7	LD_TYPE	050D
CHANS	5C4F	FINDPOS	18F9	LD_VERIF	0163
CHARS	5C36	FIXTOP	1BC4	LD_WAIT	011A
CHKBK	1840	FLAGS	5C3B	LEAV01	0FA4
CH_ADD	5C5D	FLAGS2	5C6A	LEAVER	0F99
CLOOP	180D	FLAGX	5C71	LIST_SF	5C3F
COORDS	5C7D	FN00	1B82	LODHDR	165B
CURCBN	5CBF	FN01	1B98	LOOP0	007E
CURCHL	5C51	FOPO	1B41	LOOP00	089A
DATADD	5C57	FORMAT	1AF3	LOOP01	08B0
DATMRK	0001	FRAMES	5C78	LOOP02	08B5
DEFADD	5C0B	FRBITL	1B3F	LOOPBS	189C
DEL27	1BC3	FRBITS	1B4A	MAXBNK	5CBE
DEL31	1BC2	FRCLK	1B58	MECNT	06E5
DEL35	1BC1	FRTZER	1B7E	MEM	5C68
DEL39	1BC0	GETMSG	184D	MENEWL2	0744
DEL43	1BBF	GETMXX	1858	MENEWLP	0717
DEL47	1BBE	GOTESF	16BA	MENT1	07CF
DEL51	1BBD	HEDMRK	0000	MENT2	080E
DEL55	1BBC	INDXGAP	1B10	MENT3	0825
DEST	5C4D	INITI	08E7	MENTER	0799
DF_SZ	5C6B	INPROG	5E88	MEOLD1	0724
DRVNUM	5E89	KSTATE	5C00	MEOLDLP	071C
DTEMP	5E82	K_CUR	5C5B	MEOLDV1	075A
ENDOFI	1AD8	K_DATA	5C0D	MEOLDV2	0776
ENDOFW	18EE	LAST_K	5C0B	MEOLDV3	077F
EOFMRK	0002	LDCNT	05CC	MEOLDV4	078B
EOTDELY	1BB2	LDCNT1	05DD	MEOLDVP	0752
EOTDFF	1BB4	LDCNT2	05E9	MEVARL1	078E
EOTLOOP	1AF7	LDEDG1	018D	MEVARL2	0790
ERASE0	1B2D	LDEDG2	0189	MEVARLP	0749
ERASEG	1B28	LD_8BITS	0170	MISCEL	5C7F
ERRC	5CB8	LD_BLK	05C6	MODE	5C41
ERRLN	5CB6	LD_BREAK	0111	MOTDELAR	1ABF
ERRS	5CBA	LD_BYTES	00FC	MOTDELAW	1AC5
ERRT	5CBB	LD_DAT1	0638	MOTEND	1AD6
ERR_NR	5C3A	LD_DATA	0606	MOTLOOP	1AC9
ERR_SF	5C3D	LD_DEC	016A	MOTOR	0001
ESFH00	1679	LD_DELAY	018F	MSTBOT	5CC0
ESFHEDR	1668	LD_FLAG	0159	NEWPPC	5C42

TIMEX 2068 STRINGY FLOPPY
---- SYMBOL TABLE ----

PAGE 51

NOTONE	1671	REPDEL	5C09	SKONE	1867
NOTVER	16CA	REPORT2	0293	SNOO	1A5C
NSPPC	5C44	REPORTC	0BD9	SOPO	1A1B
NUTONE	16C3	REPORTD	00F8	SRBITL	1A19
NXTLIN	5C55	REPORTR	05CA	SRBITS	1A23
OLDPPC	5C6E	REPPR	5C0A	SRCLK	1A30
ODPO	1A71	REPOO	1790	SRTZER	1A58
QSPCC	5C70	RFREAM	1789	START	5E86
PIN	5C39	RST10	0010	STKBOT	5C63
RPC	5C45	RST8	0008	STKEND	5C65
PR1	17A6	SABIT1	00BA	STOP	5C6C
PR2	17A8	SABIT2	00B7	STRLEN	5C72
PR3	17B0	SAEND	172E	STRMNM	5C0A
PR4	17B8	SALDEND	00FA	STRMS	5C10
PR5	17BD	SALDRTN	00E5	SUBPPC	5C47
PR6	17CC	SAVRYT	0068	SYNCD	0016
PR7	17D4	SAVCNT	0851	SYSCON	5C8C
PR8	17E1	SAVE	00AD	TAPE	0068
PROG	5C53	SAVEETC	01AB	TAPEND	08E7
PRTCHR	0544	SAVE_8	00CB	TVDATA	5C0E
P_RAMT	5CB4	SAV_TYPE	02B3	TV_FLAG	5C3C
QB0	1704	SAV_VNEW	02A9	T_ADDR	5C74
QB1	170D	SAV_VOLD	0295	UDG	5C7B
QB2	1714	SA_ALL	04C9	VAR8	5C4B
QB3	1719	SA_BLNK	01FB	VERFLG	5E80
QB4	1735	SA_COD1	0387	VIDMOD	5C02
QB5	173D	SA_COD2	03BC	VRCNT	058F
QB6	1745	SA_COD3	03D5	VRCNT1	05AD
QN00	19CE	SA_COD4	03FF	VRCNT2	05B8
QOPO	198D	SA_CODE	032E	VRCNT3	05C4
QRBITL	198B	SA_DAT1	02B6	WAIT	183D
QRBITS	1995	SA_DATA	0238	WAITK	08AA
QRCLK	19A2	SA_LIN1	0456	WBLOCK	1970
QRTZER	19CA	SA_LINE	0447	WORKSP	5C61
QUEND	174F	SA_LOOP	00A4	WPREAM	192F
RAMDAT	004F	SA_LOFF	00AB	WRBITL	1A6F
RAMTOP	5CB2	SA_NAME	0231	WRBITS	1A79
RASP	5C38	SA_NULL	022A	WRBYT	1A05
RB0	17F4	SA_OUT	00C2	WRCLK	1A86
RB1	17ED	SA_PARIT	00B4	WRITDATA	1919
RB2	1804	SA_SCREE	02F2	WRITEN	0002
RB3	1809	SA_SPACE	01DD	WRITEND	1926
RB4	1825	SA_TYPO	04A9	WRLOOP	196D
RB5	182D	SA_TYPE3	0440	WRTDAT	0080
RB6	1835	SEED	5C76	WRTEOF	1B15
REDBLO	175F	SKIPO	0049	WRTPRT	1ADD
REDBL1	1766	SKIP1	002C	WRTZER	1AAD
REDBLK	16F6	SKIP2	0045	XBASIC	0000
REDBYTE	17EB	SKIP3	005A	XTEMP	5E84
REDERR	1AE2	SKIP4	0076	X_PTR	5C5F